Wavelets in identification wavelets, splines, neurons, fuzzies: how good for identification

Anatoli B. Juditsky, Qinghua Zhang, Bernard Delyon, Pierre-Yves Glorennec, Albert Benveniste

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Wavelets in identification

wavelets, splines, neurons, fuzzies: how good for identification?

A. Juditsky, Q. Zhang, B. Delyon, P-Y. Glorennec, and A. Benveniste

N° 2315

September 1994

PROGRAMME 5

ISSN 0249-6399

Wavelets in identification

wavelets, splines, neurons, fuzzies: how good for identification?

A. Juditsky, Q. Zhang, B. Delyon, P-Y. Glorennec, and A. Benveniste * ** ***

Programme 5 — Traitement du signal, automatique et productique
Projet AS

Rapport de recherche n° 2315 — September 1994 — 106 pages

Abstract: This is a tutorial about nonparametric nonlinear system identification. Advantages and limitations of this approach are discussed from the engineer's point of view. Classical as well as modern techniques are discussed, this includes kernel and projection estimates, neural networks and hinging hyperplanes, and mainly wavelet estimators. Both practical and mathematical issues are investigated. Advantages and limitations of wavelet based techniques are emphasized. Finally we show how fuzzy models may play a role in this game, as a framework for expressing prior knowledge on the system. The whole material is illustrated on some application examples.

Key-words: nonparametric estimation, wavelets, neural networks, fuzzy models.

(Résumé : tsvp)

*IRISA, Campus de Beaulieu, 35042 RENNES Cedex, FRANCE, name@irisa.fr
**A.J., Q.Z., B.D., A.B. are with INRIA, P-Y. G. is with INSA
***This work has been supported in part by Alcatel-Alsthom-Recherche and European Gas Turbine SA under several contracts, C. de Maindreville, P. Durand, T. Pourchot, F. Costa, and D. Cavalerie are gratefully acknowledged.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Ondelettes et identification

Résumé : Ceci est un mini-cours en estimation non paramétrique des systèmes non linéaires. On passe en revue les estimateurs "classiques" à noyau, par projection, à splines, et aussi des techniques plus nouvelles comme les réseaux de neurones, les hyperplans de Breiman et les estimateurs à ondelettes. Ces techniques sont étudiées et discutées à la fois du point de vue de l'ingénieur et du mathématicien. On discute également de la place des modèles flous dans ce paysage, pour ce qui est de l'expression de connaissances a priori sur le système. Trois applications servent d'illustration.

Mots-clé : estimation non paramétrique, ondelettes, réseaux de neurones, logique floue.

Contents

Chapter 1

Introduction, Motivations, Basic Problems

In his inspiring tutorial [51], L. Ljung quoted the following:

An engineer, who is faced with [characterizing, or predicting, the behaviour of his plant based on recorded data] *has the following perspective:*

- *How can I best use the information in the observed data to calculate a model of the system's properties?*
- *How can I know if the model is any good, and how can I trust it for simulation and design purposes?*
- *How shall I manipulate the input signals to obtain as much information as possible about the system?*
- *What kind of software support is available for doing the tasks?*

Later on in the same article, L. Ljung discusses the question of model nature and structure. By model nature, we have in mind the following classification:

- physical models,
- semi-physical models, also called "grey-box" models,
- black-box models.

This paper mainly concentrates on the last category, namely black-box models. And, within black-box models, we shall concentrate on the less popular ones in control community, namely those that are *nonlinear and nonparametric* in nature. Here "nonlinear" means that our model class will not be restricted to linear input-output maps. And "nonparametric" means that our models do have parameters, but in a quantity that is not a priori fixed, but fully depends on data; consequently, convergence issues and quality of fit cannot be

assessed in terms of the involved parameters, but rather more globally in terms of the global behaviour. "Nonlinear and nonparametric" thus will be our general perspective throughtout this tutorial. While this setting may appear quite technical, more familiar and even some sexy ones will also be covered, such as: *neural networks [65, 42, 59], wavelets [55, 14], fuzzy models [48].* A typical form of the kind of model class that we shall consider is the popular single hidden layer neural network for static systems:

$$f_n(x) = \sum_{i=1}^{n} c_i \sigma(a_i^T x + t_i) + c_0 \ , \tag{1.1}$$

where σ is the sigmoid function, $x \in \mathbf{R}^d$ is the input, n is the number of neurons, and the (c_i, a_i, t_i)'s are the adjustable parameters. This is clearly nonlinear in x, and the size n of the network is to be tuned on the data. In addition, in this case, the model is also nonlinear in the parameters.

Such models have gained increasing interest, as reflected for instance in the articles [65, 42, 59]. This is due to their ability to encompass truly nonlinear behaviours, including those involved in classification and, more generally, decision procedures. Referring to Ljung's practical problem setting above, the following practical questions must be investigated when using nonlinear nonparametric models such as (1.1):

- *How good nonlinear nonparametric models can extrapolate or predict behaviours outside the range of data used for their identification, fitting, tuning, or training* [1]? Predicting behaviours is one of the main purposes of system identification. It is not usual to ask such a question about linear system identification, since good linear model fitting generally also provides good prediction for truly linear plants. But this is of primary concern in our case, since nonlinear systems are by essence non easily predictable outside the range of available observations. This question is also related to that of appropriate choice of inputs for identification.

- *How nonlinear nonparametric models can be used for system monitoring and diagnostics?* Such models are in principle good candidates for system monitoring, since they are able to describe systems behaviours at *all* operating points simultaneously, thus preventing from confusing between change in operating point and changes in systems behaviour. However, it is not clear how changes could be interpreted using such models, i.e., how diagnostics could be performed.

Then the user is faced with a second question, namely how identification should be performed:

- *How data should be used to fit a nonlinear nonparametric model?* Though different situations can occur, we shall mainly investigate in this paper the classical situation in which noisy input/output measurements are available.

[1] These are more or less equivalent words used by different communities, we shall use anyone of these indifferently.

- *How can one take advantage of any kind of prior knowledge for some partial or pre-tuning of the model ?* Such a coarsely tuned model is sometimes sufficient, and sometimes used as an initial guess for system identification. Also, prior information can be critical for diagnostics. Again, linear systems engineering can serve as a guide for us : response times, resonant modes, delay, and others, are typical qualitative informations that engineers may have from experience about their plants, and they know how to reflect this prior knowledge into linear models. For nonlinear nonparametric models, no obvious alternative seems to exist : what kind of prior knowledge is relevant for such models, and how to express it ? Thus it seems that the engineer must entirely rely on fitting from data, without taking advantage of some prior knowledge he may have ; we shall see that fuzzy models and their rules may be good candidates to express such prior knowledge.
- *What kind of software support is available for doing the tasks ?*

Moving one step further toward mathematical formulation of our problems, we may translate some of the above questions into the more technical following ones :

- *How to assess the quality of approximation ?* Given a true system f, and an approximation $\widehat{f}$ of it, how to measure the quality of approximation ? No parametric distance can be used. And since nonlinear systems are considered, usual operator norms from linear system theory cannot be considered. In the second part of this chapter, based on a few examples, we shall introduce the distance measures we shall use throughout this paper. These will mainly be L_p-type norms involving $f - \widehat{f}$ and possibly some derivatives of it. Note that using such distance measures involves some kind of prior knowledge, namely the assumption that the system in consideration belongs to the considered space, and this is a smoothness prior information.
- *How to measure the quality of fit from noisy data ?* This is really assessing the quality of system identification. We shall naturally use figures of merit of the form $\mathbf{E}\|f - \widehat{f}_N\|$, where $\|.\|$ denotes a norm such as discussed before, $\widehat{f}_N$ is the estimate of f based on an N-sample record, and $\mathbf{E}$ is the expectation with respect to all kinds of uncertainties (input, output, and noise).
- *What plays the role of "Cramer-Rao bounds", and what means for an estimator to be "optimal" ?* Such criteria are important in assessing relative performance of estimators, especially because of the very large variety of the models and identification procedures proposed so far.
- *How efficient identification algorithms really are in terms of computational cost and quality of conditioning ?* Since our model classes often are nonlinear in the parameters, tuning procedures may be of prohibitive cost and may further be illbehaved (cf. the wellknown "backpropagation" algorithm for neural network training).
- *What kind of coarse or qualitative property can be asserted about the models we consider,* apart from smoothness prior information such as discussed before ?

These are some of the issues that we shall discuss throughout this tutorial. The paper is organized as follows. The remainder of this chapter is devoted to the two applications we selected for a more detailed discussion. Then we discuss some basic mathematical problems relevant to our nonparametric setting, and justify by the way the use of some specific distance measures between systems and their estimates.

In Chapter 2 the classical background of nonparametric estimation is visited. First, so-called "linear" estimators (be careful that systems and models are nevertheless nonlinear) are presented and discussed : Kernel, piecewise polynomial, and projection estimators are typical instances. Then the issue of selecting the "model order" is discussed and Generalized Cross Validation is introduced. In a second section, convergence rates and performance criteria are analysed, and it is shown that classical linear estimators perform poorly for systems with sparse singularities — such nonlinear systems frequently occur in practice. Some existing nonlinear estimation techniques which provide spatial adaptation are briefly discussed in the last section, these include sigmoid based neural networks, and an interesting alternative proposed by Leo Breiman, namely the "hinging hyperplanes" which are in fact piecewise linear models such as used in control by E.D. Sontag in the early eighties [73]. Such nonlinear estimators with spatial adaptation are not supported by satisfactory mathematical analysis, however. This motivates investigating wavelets.

Wavelets are introduced in Chapter 3 and their contribution to function approximation theory is briefly reported. In particular, orthonormal bases of wavelets for L^2-type spaces is presented. The importance of *Besov spaces* of functions is emphasized, for modelling smooth systems with sparse singularities. Besov spaces are closely related to the more usual Sobolev spaces. The optimality of wavelet expansions in Besov spaces of functions is discussed. The central role of Besov spaces for wavelets was pointed out by Yves Meyer.

How wavelets can be best used for estimation is the topic of Chapter 4. We report on and discuss the simple and elegant method of "wavelet shrinking" as introduced by David Donoho and co-workers.

Building orthonormal bases of wavelets, for even medium large dimensional input spaces (say, $\geq$ 10), becomes prohibitive in terms of memory requirements. Thus an alternative method is proposed in Chapter 5, which is still based on wavelets, but in a different manner. This method is suitable for sparse training data sets, i.e., data sets whose cardinality does not grow exponentially with the dimension of the input space.

Now, the question of how to practically express available prior knowledge for nonparametric models is still open. In Chapter 6 we discuss a proposal toward achieving this, which is based on fuzzy models and their associated rules. An extension of the usual fuzzy models is proposed to capture multiresolution aspects of wavelet based estimators.

Experimental results of some of these methods are reported in Chapter 7.

Finally, both practical and mathematical aspects are summarized and discussed in the conclusion.

1.1 Two application examples

1.1.1 Modelling a gas turbine system, an example of identification of a static nonlinear system

In this subsection we briefly present the case study of a gas turbine system, as an example of identification of a static nonlinear system. Results and experiments will be reported in chapter 7. Gas turbines are power motors, typically used in electrical power generators and aircrafts. Usually a gas turbine system is mainly composed of a compressor, one or several combustion chambers and an expansion turbine, as illustrated in figure 1.1. The compressor produces high pressure air which is then mixed with the fuel. This mixed gas is burned in the combustion chambers to increase its temperature and pressure. The burned gas is then forwarded to the expansion turbine. The pressure of the gas drives the rotor of the expansion turbine, which in turns drives the compressor. The residual energy can then be used for producing electricity, and the gas is rejected at the exhaust of the expansion turbine.

One of the purposes of our joint study with European Gas Turbine SA, Belfort, and Alcatel-Alsthom-Recherche, Marcoussis, was to develop a monitoring and diagnostics system for the joint system {combustion chambers, expansion turbine}. Monitoring is based on the measured pressure in the compressor, the rotation velocity of the turbine and measurements from the thermocouples available at the exhaust of the expansion turbine. Thus no direct observation is available on the status of the combustion chambers, see figure 1.1. Hence a semi-physical model has been developed that predicts the profile of temperature at the exhaust of the expansion turbine using the pressure in the compressor, the mean temperature at the exhaust of the expansion turbine, and the rotation velocity of the turbine [86, 90]. This model consists of two parts : first the unknown temperature profile within the chambers is modelled as a linear regression involving one parameter per chamber ; then, based on basic thermodynamics, a relation between this profile and the temperature profile at the exhaust of the expansion turbine is given. Since the gas flow rotates within the turbine during its expansion, a phase shift between the two input and output temperature profiles is exhibited. Therefore, some phase shift parameter appears in the model which makes it strongly nonlinear. This model is semi-physical and inaccurate since the input temperature profile uses as a regression function some waveform based on qualitative knowledge, and very simplified thermodynamics is used for gas diffusion in the expansion turbine.

This semi-physical modelling was for the purpose of monitoring the turbine system. Despite its inaccurate nature, the model has been successfully used for developing a monitoring system of the combustion chambers, see [90]. Unfortunately, this model is not entirely satisfactory for some other purposes, such as the monitoring of the thermocouples installed at the exhaust of the expansion turbine. The purpose of this discussion is to compare results from this semi-physical model with some alternative nonparametric identification method based on wavelets, and discuss the two questions of respective accuracy of fit and explicative power of these two styles of models.

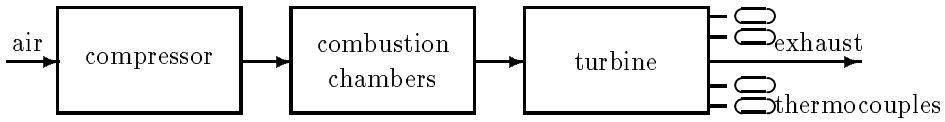

Figure 1.1: A gas turbine system

1.1.2 Modelling the hydraulic actuator of a robot arm, an example of identification of a dynamic nonlinear system

Now let us consider the modelling of the actuator of a robot arm [2]. It is a hydraulically driven arm. By controlling the position of a valve, the oil pressure in the transmission circuit is regulated. The oil pressure drives the motion of the arm. What we want to model is the relationship between the position of the valve and the oil pressure, both quantities being measured. In fact the valve directly regulates the oil streams injected in the transmission circuit. Hence variation of the oil pressure depends not only on the position of the valve, but also on the quantity of the oil accumulated in the transmission circuit, which in turn is reflected by the oil pressure. Clearly this is a dynamic system : variation of its output (oil pressure) depends on both its input (the position of the valve) and its state (reflected by the oil pressure). We tried to model this dynamic system with linear ARX models, but the results were not satisfactory. Therefore, we decided to apply some nonlinear nonparametric model and see if we can improve the performance of the modelling.

1.1.3 Prediction of glycæmic variations, an example of identification of a dynamic nonlinear sytem with imprecise and incomplete data.

Glycæmic variations depend on several factors which are not easily quantifiable and, moreover, may vary with time. Food diet, physical activity, stress and emotions, proximity of meal, have effects that the doctors know how to *qualitatively* assess. For a healthy person, glycæmic regulation is ensured via the secretion of insulin by the pancreas. In case of organic deficiency, for diabetic persons, insulin must be artificially injected. Deciding the amount for injection is very difficult, because morphology, future physical activity, time of meal, glucide richness of meal, present glucose concentration, and results of the previous day, have to be taken into account. Moreover, injected insulin acts with delay, and its efficiency reduces as glucose concentration gets higher. Lastly, hypoglycæmia is almost allways followed by hyperglycæmia. For an optimum glycæmic control, it would be better to anticipate before the glucose level rises, as it occurs for endogenic insulin secretion in healthy persons. To summarize, we have to deal with a nonlinear, unstable system, with time delay.

[2] This application has been borrowed from Linköping University, while Q. Zhang was visitor at the Automatic Control group.

Doctors have devised empirical rules allowing the diabetic persons to approximatively compute themselves the insulin level for injection. For diabetic persons using a pump, insulin injection rate has two parts : the basic flow rate, denoted $B_a(t)$, and providing about 50% of daily insulin needs, and a variable part, the bolus, denoted $B_o(t)$, which is a flash injection to assimilate a recent meal.

Nevertheless, despite doctor's experience, it is very difficult to manually obtain a more or less constant glycæmic level, in part because a good control should take into account up to six input variables, which is far beyond human control capability. This motivated us to propose a predictive glycæmic model, as a basis for automatic injection control. This model uses as a basis the empirical rules of doctors, and takes into account the qualitative nature of available data. For this proposal, we have several "self-surpervision note-books", i.e., daily support to control the context and the treatment of insulin-dependent diabetic patients under pump operation. Thus, each day the diabetic writes on his note-book 1/ time and actual glycemia, 2/ time, importance and quality of his meal, 3/ activity, 4/ insulin injection. Experimental results on this case study are reported in Section 7.3.

1.2 Basic mathematical problems

Here we establish the general framework of *nonparametric regression* we shall use throughout this article, and we justify the use of particular distance measures between true system and its estimate we shall deal with in the sequel.

Problem 1 (nonparametric regression) *Let (X, Y) be a pair of random variables with values in $\mathcal{X} = \mathbf{R}^d$ and $\mathcal{Y} = \mathbf{R}$ respectively. A function $f : \mathcal{X} \mapsto \mathcal{Y}$ is said to be the* **regression function of Y on X** *if*

$$\mathbf{E}(Y|X) = f(X) \ . \tag{1.2}$$

A typical case is $Y = f(X) + e$, where e is zero mean and independent of X. For $N \geq 1$, $\widehat{f}_N$ shall denote an estimator of f based on the random sample $\mathcal{O}_1^N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ of size N from the distribution of (X, Y), i.e., a map

$$\widehat{f}_N \quad : \quad \mathcal{O}_1^N \mapsto \widehat{f}_N\left(\mathcal{O}_1^N, .\right) \tag{1.3}$$

where, for fixed $\mathcal{O}_1^N$, $x \mapsto \widehat{f}_N\left(\mathcal{O}_1^N, x\right)$ is an estimate of the regression function $f(x)$. The family of estimators $\widehat{f}_N$, $N \geq 1$ is said to be **parametric** *if $\widehat{f}_N \in F$ for all $N \geq 1$, where F is some set of functions which are defined in terms of a* fixed *number of unknown parameters. Otherwise the family of estimators $\widehat{f}_N$, $N \geq 1$ is said to be* **nonparametric.**

For the sake of convenience, we shall often refer to X and Y as the *input* and *output* respectively (although they do not need to be such in actual applications). Our objective in this section is to give a short overview of some basic instances of nonparametric regression. Two typical problems are considered in the statistical litterature, namely the

- *nonparametric regression with random design (or sampling),* where it is assumed that the variables X_i are random, independent, and identically distributed on $[0,1]^d$ with density $g(x)$, and the
- *nonparametric regression with deterministic design (or sampling),* where it is assumed that the input variables X_i are nonrandom; the simplest case of deterministic design is the *regular design,* where the inputs X_i form a regular grid (for instance, $f : \mathbf{R} \to \mathbf{R}$ and $X_i = i/N$).

In the remainder of this chapter we consider the random design only, although the observations (X_i, Y_i) are allowed to be dependent.

Nonparametric Regression for Static Systems. This is the simplest case. The considered system has the form

$$Y_i = f(X_i) + e_i, \qquad i = 1, \ldots, N \ , \tag{1.4}$$

where $f(x) : \mathbf{R}^d \mapsto \mathbf{R}$, and, for the sake of simplicity, we assume that e_i are independent Gaussian random variables with $\mathbf{E}e_i = 0$ and $\mathbf{E}e_i^2 = \sigma_e^2$.

Adaptive classification and density estimation[3]. The problem of classification (discriminant analysis, or statistical pattern recognition) is usually formulated as follows. Let X be a random variable with values in $\mathbf{R}^d$, and let the label Z denote a random variable which takes values in some finite set $\mathcal{Z} = \{z_1, \ldots, z_M\}$; the symbol z shall denote a generic element of this finite set. We want to guess the value of Z when X is observed. We consider the case in which the random vector X has probability density $f(x)$ and conditional densities $f(x|z)$ given that $Z = z$, the general case is handled similarly. We call a *solution* any measurable function $g : \mathcal{X} \mapsto \mathcal{Z}$, and $\mathbf{P}(g(X) \neq Z)$ is the corresponding *error probability.* The distribution of the pair (X, Z) is defined by the distribution μ of X and the regression functions

$$\mathbf{p}(z|x) = \mathbf{P}(Z = z|X = x) = \frac{\mathbf{p}(z)\ f(x|z)}{f(x)}, \quad x \in \mathbf{R}^d \, ,$$

where Bayes' rule has bee used for the second equality, and $\mathbf{p}(z) = \mathbf{P}(Z = z)$. Functions $f(x|z)$ are also called *a posteriori densities.* The solution $g^*(x)$ is called *Bayesian* or *Maximum A Posteriori — MAP,* if

$$\mathbf{p}(g^*(x))\ f(x|g^*(x)) = \max_z \mathbf{p}(z) f(x|z) \quad \text{a.e. } x \ . \tag{1.5}$$

The Bayesian solution g^* minimizes the error probability, i.e.,

$$\mathcal{L}^* \triangleq \min_g \mathbf{P}(g(X) \neq Z) = \mathbf{P}(g^*(X) \neq Z) \ , \tag{1.6}$$

[3] in this section we follow the presentation of ch. 10 in [21]

and $\mathcal{L}^*$ is called the *Bayesian error probability.*

In *adaptive classification,* we want to minimize the error probability when the true $\mathbf{p}(z)$ and $f(x|z)$ are unknown and a training sample $\mathcal{O}_1^N = \{(X_1, Z_1), \ldots, (X_N, Z_N)\}$ of N independent observations distributed as (X, Z) is available. We assume that the training sample $\mathcal{O}_1^N$ and the test sample (X, Z) are independent. The estimate $g_N(X)$ of Z is now a measurable function of X and $\mathcal{O}_1^N$, and the following *conditional error probability* is a quantity of interest :

$$\mathcal{L}_N = \mathbf{P}(g_N(X) \neq Z \mid \mathcal{O}_1^N) \ . \tag{1.7}$$

In particular, we search for a sequence of estimates g_N such that

$$\mathcal{L}_N \to \mathcal{L}^* \quad \text{almost surely.} \tag{1.8}$$

Referring to (1.5), the Bayesian solution can be approximated by the function g_N characterized by

$$\widehat{\mathbf{p}}(g_N(x))\widehat{f}(x \mid g_N(x)) = \max_z \widehat{\mathbf{p}}(z)\widehat{f}(x|z) \ . \tag{1.9}$$

where $\widehat{f}(.|z)$ are estimates of $f(.|z)$ based on $\mathcal{O}_N$. There is a simple way to measure the conditional error probability $\mathcal{L}_N$ for the adaptive classifiers which satisfy (1.9) : Devroye and Györfi, [21] have shown that, if the random vector X is distributed with some density f and g_N is defined via (1.9), then

$$0 \leq \mathcal{L}_N - \mathcal{L}^* \leq \sum_{z \in \mathcal{Z}} \int |\mathbf{p}(z)f(x|z) - \widehat{\mathbf{p}}(z)\widehat{f}(x|z)| \ dx.$$

Different versions of this result were proved in [79], [83], [13], among others. This result implies that the classification error can be bound using the L_1 norm[4] of the estimation error of the density $\mathbf{p}(z)f(x|z)$. Thus we have related the problem of adaptive classification to that of estimating the density of a random variable in L_1-norm. Other advantages of considering the averaged L_1-norm are discussed in [21]. Alternative distance measures for densities are often considered, e.g., averaged L_2-norm (often used, since it seems to be the easiest to estimate) or L_∞-norm.

Nonparametric Regression with Dynamics. Consider the following dynamical system :

$$Y_i = f(\Phi_i) + e_i, \qquad i = 1, \ldots, N \ ,$$

where $Y_i \in \mathbf{R}$ and $\Phi_i \in \mathbf{R}^d$ are observed, and e_i is a white noise as above. We assume that

$$\Phi_i = (Y_{i-1}, \ldots, Y_{i-m}; U_i, \ldots, U_{i-p}) \ , \tag{1.10}$$

[4] Recall that for a function $g : \mathbf{R}^d \to \mathbf{R}$ the L_p norm is defined for $0 < p < \infty$: $\|g\|_p = (\int |g(x)|^p dx)^{1/p}$, and for $p = \infty$: $\|g\|_\infty = \text{ess sup}_x |g(x)|$.

where $U_i \in \mathbf{R}$ denote the inputs ($m + p = d$). For example, if $\Phi_i = (Y_{i-1}, \ldots, Y_{i-d})$, then

$$Y_i = f(Y_{i-1}, \ldots, Y_{i-d}) + e_i \ . \tag{1.11}$$

In analogy with the corresponding parametric model we call this system a *nonparametric autoregression* or a *functional autoregression* of dimension d (FAR(d)). As an interesting application, we can consider a simple controlled FAR model for adaptive control :

$$Y_i = f(\Phi_i) + U_i + e_i \ , \tag{1.12}$$

where $\Phi_i = (Y_{i-1}, \ldots, Y_{i-m})$, and U_i is the control. The following question can be considered : how to choose the control (U_i) for the system (1.12) to track some reference trajectory $y = (y_i)$, or, at least, how to choose U_i in order to minimize $\mathbf{E}Y_i^2$, or, simply, to stabilize the system (1.12) ? If the function $f(\Phi)$ was known, we could use control

$$U_i = -f(\Phi_i)$$

to obtain $Y_i = e_i$. Clearly, this is a "minimum variance" control, since $\mathbf{E}Y_i^2 \geq \sigma_e^2 = \mathbf{E}e_i^2$. If f is unknown, a possible solution consists in performing nonparametric "certainty equivalence control": compute an estimate $\widehat{f}_N$ of the regression function f based on the observations of the input/output pair $(\Phi_i, Y_i - U_i)$, and then take

$$U_i = -\widehat{f}_i(\Phi_i) \ . \tag{1.13}$$

To analyse the certainty equivalence control (1.13), let us consider the control cost

$$Q_N \ = \ \frac{1}{N}\sum_{i=1}^{N} Y_i^2 \ = \ \frac{1}{N}\sum_{i=1}^{N}(f(\Phi_i) - \widehat{f}_i(\Phi_i))^2 + \frac{1}{N}\sum_{i=1}^{N} e_i^2 \ .$$

It is easily checked that

$$\mathbf{E}(\widehat{f}_i(\Phi_i) - f(\Phi_i))^2 \to 0 \ \text{ when } i \to \infty \tag{1.14}$$

implies $\mathbf{E}Q_N \to \sigma_e^2$, and $\widehat{f}_i(\Phi_i) - f(\Phi_i) \to 0$ a.e. implies $Q_N \to \sigma_e^2$ a.e. Thus condition (1.14) is instrumental in analysing this problem, and we shall informally discuss how it can be guaranteed.

Denote by $\Phi_0^{i-1} = (\Phi_0, \ldots, \Phi_{i-1})^T$ the vector of all available inputs up to time $i-1$, and by $\varphi_0^{i-1} = (\varphi_0, \ldots \varphi_{i-1})^T$ the corresponding vector of integration variables. Let $\mathbf{P}$ denote the distribution of the vector sequence (Φ_i) when driven by the unknown "true" model (1.12)–(1.13), let $\mathbf{P}_{\Phi_0^{i-1}}(\cdot)$ be a distribution of Φ_0^{i-1}, and let $\mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(\cdot)$ be a conditional density of the distribution of Φ_i given Φ_0^{i-1} (we assume that such a density exists). We have

$$\mathbf{E}|\widehat{f}_i(\Phi_i) - f(\Phi_i)|^2 \sim \int |\widehat{f}_i(x) - f(x)|^2 \ \mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(x)\ dx\ \mathbf{P}_{\Phi_0^{i-1}}(d\varphi_0^{i-1}) \ .$$

Note that, if the closed-loop system (1.12)–(1.13) is stable, one would reasonably take equal weights for the observations $\Phi_0, \ldots, \Phi_i$ in the estimate $\widehat{f}_i$. In such a case the estimate $\widehat{f}_i(\Phi)$

is asymptotically (as $i \to \infty$) slowly varying, i.e., $\widehat{f}_i \sim \widehat{f}_{i-1}$. Thus we can write informally

$$\mathbf{E}|\widehat{f}_i(\Phi_i) - f(\Phi_i)|^2 \sim \int \mathbf{P}_{\Phi_0^{i-1}}(d\varphi_0^{i-1}) \int |\widehat{f}_{i-1}(x) - f(x)|^2 \, \mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(x) \, dx \ .$$

The latter integral can be bound in several ways. For instance,

$$\begin{aligned} \int |\widehat{f}_{i-1}(x) - f(x)|^2 \, \mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(x) dx &\leq \sup_x \ |\widehat{f}_{i-1}(x) - f(x)|^2 \int \mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(x) \, dx \\ &= \sup_x \ |\widehat{f}_{i-1}(x) - f(x)|^2, \end{aligned}$$

which yields the bound

$$\mathbf{E}|\widehat{f}_i(\Phi_i) - f(\Phi_i)|^2 \leq \mathbf{E} \sup_x |\widehat{f}_{i-1}(x) - f(x)|^2 = \mathbf{E}\|\widehat{f}_{i-1} - f\|_\infty^2 \ .$$

On the other hand, if the conditional density is bounded, i.e., $\mathbf{p}_{\Phi_i|\Phi_0^{i-1}} \leq C_p$, then

$$\int |\widehat{f}_{i-1}(x) - f(x)|^2 \, \mathbf{p}_{\Phi_i|\Phi_0^{i-1}}(x) \, dx \ \leq \ C_p \int |\widehat{f}_{i-1}(x) - f(x)|^2 \, dx = C_p \|\widehat{f}_{i-1} - f\|_2^2 \ .$$

Thus, as a conclusion, in any case, the crux in analysing this adaptive minimum variance nonlinear control consists in getting bounds for the error in estimating the unknown function f. Hence, in addition to proving consistency for the estimates, getting such bounds is an important question.

Discussion. This section about basic mathematical issues can be summarized as follows:

1. Nonparametric estimation of regression functions is instrumental in various problems such as adaptive identification, classification, and control.
2. Averaged L_p-norms of estimation error for various p's are natural candidates as a figure of merit. We shall see later that error measures involving also derivatives of f and $\widehat{f}$ will be useful, so that smoothness of estimates can also be guaranteed.
3. Having bounds for the estimation error is of paramount importance. This has been illustrated on the adaptive control example. But we shall see later that some estimators can exhibit arbitrarily poor performance for some "bad" systems, so that having error bounds is really needed to prevent the user from getting bad results.

Chapter 2

"Classical" methods of nonlinear system identification

Throughout this chapter, Problem 1 is considered. We first discuss some estimators that are *linear,* i.e., that satisfy $\widehat{f+g} = \widehat{f} + \widehat{g}$; note that functions f, g, and their estimates, are generally nonlinear as functions of their input x. Linear estimators build the folklore of nonparametric estimation: kernel estimators, projections on linear subspaces of functions, are typical instances we shall describe. We shall then discuss, both practically and theoretically, some severe practical limitations of linear estimators. Roughly speaking, linear estimators are suitable for systems with "uniform smoothness" ; systems with sparse singularities (e.g., hard limiters, quantizers, some mechanical systems) are poorly handled. This motivates the search for new nonlinear estimators, neural networks, and some related methods, are candidates we shall briefly scan.

2.1 Linear Nonparametric Estimators

All estimators presented in this subsection are linear ones, i.e. they have a common general form

$$\widehat{f}_N(x) = \sum_{i=1}^{N} Y_i \; W_{N,i}(x) \; , \quad W_{N,i}(x) = W_{N,i}(x, X_1, \ldots, X_N) \tag{2.1}$$

where we recall that $\mathcal{O}_1^N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ is the given random sample observation, and the weights $W_{N,i}(x)$ only may differ.

2.1.1 Some Linear Nonparametric Estimators

Kernel estimators for regression functions and densities

Kernel estimators were first proposed by Nadaraya and Watson in 1964 [58] and [82]. The Nadaraya-Watson kernel estimator is an interpolation procedure. It is given by

$$\widehat{f}_N(x) = \frac{\sum_{i=1}^N Y_i \; K\left(\frac{x-X_i}{h_N}\right)}{\sum_{i=1}^N K\left(\frac{x-X_i}{h_N}\right)} \; , \tag{2.2}$$

where (h_N) is a sequence of positive numbers, $h_N \to 0$ as $N \to \infty$, and K is a function on $\mathbf{R}$ satisfying

$$\lim_{|u|\to\infty} |u||K(u)| = 0, \quad \int_{-\infty}^{\infty} |K(u)|du < \infty, \quad \sup_{u\in\mathbf{R}} |K(u)| < \infty, \quad \int_{-\infty}^{\infty} K(u) = 1 \; . \tag{2.3}$$

The positive number h_N is called the *bandwidth* and the function K satisfying (2.3) is called a *kernel*; in fact, h_N is better interpreted as a scaling factor. Clearly, the Nadaraya-Watson estimator is linear, and has the form (2.1). Typical examples of kernels are $K(u) = (1/2)\, 1_{\{|u|\le 1\}}$ (rectangular window kernel), and $K(u) = (1/\sqrt{2\pi}) \exp(-|u|^2/2)$ (Gaussian kernel), etc... Usually K is chosen to be an even function.

The idea of kernel estimation is simple, let us explain it for the case of the rectangular kernel in one dimension. In this case the estimator (2.2) is a simple moving average with equal weights: the estimate at point x is the average of observations Y_i corresponding to X_i's belonging to the "window" $[x - h_N, x + h_N]$. If $h_N \to \infty$ then the estimator tends to $N^{-1}\sum_i Y_i$, the average of all observations, and thus for functions f which are far from being constant, the bias becomes large. If h_N is very small (say, smaller than the pairwise distance between sample points X_i) then the estimator reproduces the observations: $\widehat{f}_N = Y_i$. In this extremal case the variance of the error becomes high. Thus increasing h_N tends to increase the bias of estimator, while reducing h_N leads to a larger variance. The optimal choice for h_N corresponds to an equal balance between bias and variance.

Also closely related to estimator (2.2) is the Parzen-Rosenblatt kernel estimator for densities. Let $X_1, \ldots, X_N$ be independent and identically distributed random variables with common density $f(x), x \in \mathbf{R}^d$. The Parzen-Rosenblatt estimator of density $f(x)$ is a suitably smoothed histogram. It is defined as [62], [70]

$$\widehat{f}_N(x) = \frac{1}{N h_N^d} \sum_{i=1}^N K\left(\frac{x - X_i}{h_N}\right) \; , \tag{2.4}$$

where d is the state-space dimension of X and K is a kernel as in (2.3). Kernel estimate (2.2) can be easily derived from the Parzen-Rosenblatt one. Recall definition (1.2) of the regression function, take the Parzen-Rosenblatt estimator (2.4) for the joint density $f(x, y)$

of (X, Y) and denote it by $\widehat{f}_N(x, y)$. Then, replacing, in formula

$$f(x) = \frac{\int y\ f(x,y)\ dy}{\int f(x,y)\ dy}\ ,$$

$f(x)$ and $f(x, y)$ by their corresponding Parzen-Rosenblatt estimates, yields Kernel estimate (2.2).

We now state a sample of results about the properties of kernel estimates for the d-dimensional case. Assume that it is known a priori that f belongs to the ball $\mathcal{C}^s(L)$ in the so-called Hölder space : for s and L positive, let $\mathcal{C}^s(L)$ be the family of functions $f(x),\ x \in [0,1]^d$ defined by [1]

$$\mathcal{C}^s(L) = \left\{ f :\ \ |f^{(k)}(x) - f^{(k)}(x')| \leq L|x - x'|^{s-k},\ \text{for any } x, x' \in [0,1]^d \right\}\ \ , k = \lfloor s \rfloor\ . \quad (2.5)$$

Note that this is a smoothness prior of the kind we discussed in our introduction. If $s \geq 1$ is integer, then $\mathcal{C}^s(L)$ contains continuous functions having Lipschitz $(s-1)$-th derivative. We can now give a result on the rate of convergence of the kernel estimate. We acknowledge Rosenblatt [71] for the first two statements of it, though it probably belongs to the earlier folklore of nonparametric statistics.

Theorem 1 ([71]) *Let $\widehat{f}_N$ be a kernel estimate with bandwidth h_N such that $h_N \to 0$ and $Nh_N \to \infty$, with kernel K satisfying $\int x^j K(x)dx = 0$ for $j = 1, \ldots, k$. Here, x^j denotes any product of the form $x_1^{j_1} x_2^{j_2} \ldots x_d^{j_d}$ where $j_1 + j_2 + \ldots + j_d = j$, and $x_1, \ldots, x_d$ are the coordinates of x. Assume that the observations X_i are independent and identically distributed on $[0,1]^d$ with density $g(x) \geq c > 0$, $g \in \mathcal{C}^s(L)$, and that the noise satisfies $\mathbf{E}e_i = 0$ and $\mathbf{E}e_i^2 \leq \sigma_e^2 < \infty$. Then*

1. *Uniformly over $f \in \mathcal{C}^s(L)$ and $x \in [0,1]^d$, we have the pointwise bound*

$$\mathbf{E}|\widehat{f}_N(x) - f(x)|^2 \leq C\left(L^2 h_N^{2s} + \frac{\sigma_e^2}{Nh_N^d}\right)\ . \quad (2.6)$$

The optimal value of h_N which minimizes the right-hand side of (2.6) is given by

$$h_N = \left(\frac{\sigma_e^2}{L^2 N}\right)^{1/(2s+d)}\ . \quad (2.7)$$

For this value of h_N

$$\mathbf{E}|\widehat{f}_N(x) - f(x)|^2 \leq CL^{2/(d+2s)}\left(\frac{\sigma_e^2}{N}\right)^{2s/(2s+d)}\ .$$

2. *If we consider instead the global error measure $\mathbf{E}||\widehat{f}_N - f||_2^2$, using again the same optimal value (2.7) for h_N yields the same bound, uniformly over $f \in \mathcal{C}^s(L)$.*

[1] $\lfloor s \rfloor$ denotes the maximal integer $k < s$.

COMMENTS :

1. As expected from the above informal discussion concerning the rectangular kernel, the bound for the estimation error variance given on the right hand side of (2.6) is decomposed into *bias* + *variance* terms. And, as expected, the optimal choice of h_N in (2.7) exactly balances these two terms.

2. Note that we have both pointwise and global bounds, which reflects the local nature of kernel estimates.

3. The properties of the Parzen-Rosenblatt algorithm of density estimation are identical when the unknown density f satisfies $f \in \mathcal{C}^s(L)$. Note that, since supp $f \subseteq [0,1]^d$, the L_1-norm of the error (restricted to the $[0,1]^d$) is dominated by the L_2-norm. So we get from the second statement of the theorem

$$\mathbf{E}\|\widehat{f}_N - f\|_1^2 \leq C L^{2/(d+2s)} \left(\frac{\sigma_e^2}{N}\right)^{2s/(2s+1)}$$

   provided h_N is chosen as in (2.7).

4. Often the following recursive version of the kernel estimator is considered, [33], [61] :

$$\widehat{f}_n(x) = \Gamma_n^{-1}(x) \left(\sum_{i=0}^{n} Y_i h_i^{-d} K\left(\frac{x - X_i}{h_i}\right)\right) \quad \text{if } \Gamma_n(x) \neq 0 \text{ and } \widehat{f}_n(x) = 0 \text{ if } \Gamma_n(x) = 0 \ ,$$

$$\Gamma_n(x) = \sum_{i=0}^{n} h_i^{-d} K\left(\frac{x - X_i}{h_i}\right) \ ,$$

   or

$$\widehat{f}_n(x) = \widehat{f}_{n-1}(x) + \Gamma_n^{-1}(x) \left(Y_n - h_n^{-d} K\left(\frac{x - X_n}{h_i}\right) \widehat{f}_{n-1} \right) \ ,$$

$$\Gamma_n(x) = \Gamma_{n-1}(x) + h_n^{-d} K\left(\frac{x - X_n}{h_n}\right) \ . \tag{2.8}$$

   In this form the algorithm resembles very much the recursive Least Squares algorithm for estimating the parameters of linear models. When the bandwith is such that $h_i = h\ i^{-\alpha}$ for some $0 < \alpha < 1$, the properties of the algorithm (2.8) in the static regression problem are essentially the same as those of the "off-line version" (2.2). In [61], [67] and [33] this algorithm was used to identify stable nonparametric autoregression models of the form (1.11), and the convergence of this estimator was proved. Furthermore, the same algorithm was used to provide the estimates of $\widehat{f}_n$ in the closed loop system (1.12)–(1.13), and the stability of such an adaptive control scheme was proved — [61] and [67] consider essentially one-dimensional case, and in [33] the general multi-dimensional case is studied.

Piecewise-polynomial estimators

Another nonparametric regression estimator which is commonly used is the piecewise-polynomial one. The idea is the same as for the kernel estimator, though the averaging is made over *bins* (i.e., small cubes) of fixed size δ_N rather than in h_N-neighborhood of the current point x. It is also closely related to the *radial-basis function (RBF) networks* with rigid location for the radial functions, see [65], [80]. The simplest example of this method is the piecewise-constant estimator or *regressogram*. The value of the estimate $\widehat{f}_N$ in each bin equals the average of observations Y_i such that corresponding X_i belong to the bin. For the sake of clarity we consider one-dimensional case.

The piecewise-polynomial estimator can be formally defined in terms of the following optimization problem. Let $\delta_N \to 0$ be a positive sequence, and we assume that $\delta_N^{-1} = M$ is an integer. Define $u_l = l\delta_N$, $l = 0, \ldots, M$, and divide the interval $[0,1]$ into M cubes (bins) of the form $U_1 = [0, u_1)$, $U_2 = [u_1, u_2)$, $\ldots$, $U_M = [u_{M-1}, 1]$, so each bin has length δ_N. Set $F(x) = (1, x, \frac{x^2}{2}, \ldots, \frac{x^k}{k!})^T$ and, for each bin $U_l, l = 1, \ldots, M$, solve for $\theta \in \mathbf{R}^{k+1}$ in the least squares sense the system of equations

$$Y_i = \theta^T F\left(\frac{X_i - u_{l-1}}{\delta_N}\right), \ X_i \in U_l \tag{2.9}$$

and denote by $\widehat{\theta}_{N,l}$ the corresponding solution. Then the piecewise-polynomial estimate $\widehat{f}_N$ of order k in each bin U_l is expressed as

$$\widehat{f}_N(x) = \widehat{\theta}_{N,l}^T \ F\left(\frac{x - u_{l-1}}{\delta_N}\right), \quad x \in U_l \tag{2.10}$$

The value δ_N is called the *binwidth*. As for the bandwidth h_N of kernel estimate, the binwidth tunes the smoothness: larger δ_N leads to a higher bias, and smaller δ_N results in a higher variance. In order for the least-squares problem in (2.10) to be nondegenerate we require that the number of points X_i in each bin is larger than $k+1$.

Stone, [74] has proved a result similar to theorem 1 for this type of estimate (see (2.5) for the definition of the Hölder space $\mathcal{C}^s(L)$). We state this result in the general d-dimensional case. Assume that the observations X_i satisfy the assumptions of theorem 1. Let $\widehat{f}_N$ be a piecewise polynomial estimate of order $k = \lfloor s \rfloor$, with binwidth $\delta_N \to 0$ and $N\delta_N \to \infty$ as $N \to \infty$. Then *statement 1 of theorem 1 holds with binwidth δ_N substituted for the bandwidth h_N.*

Comments :

1. Note that, unlike for Kernel estimates, piecewise polynomial estimates compute projections on the fixed set of functions $F\left(\frac{x-u_{l-1}}{\delta_N}\right), x \in U_l$ (the l-th bin). The same remark holds for the projection estimate to follow.
2. As can be seen, piecewise polynomial and kernel estimates have the same asymptotic accuracy when $N \to \infty$.

3. If f is a smooth function (i.e., $s \geq 1$), the optimal number of bins is $n_\delta \sim \delta_M^{-1}$ and is much less than the number of observations ($n_\delta \sim N^{1/3}$ for $s = 1$). This number is equivalent to the memory size required to implement the algorithm: to reconstruct the estimate, $k = \lfloor s \rfloor$ coefficients are necessary. Thus, if N is large, this algorithm offers significant advantage, in terms of memory requirements, over kernel estimates in which all measurements should be kept to reconstruct $f(x)$. Also, computing (2.9)–(2.10) is of lower computational burden than computing (2.2). These two points make the piecewise polynomial estimate more attractive.

4. Unfortunately there is no reasonable recursive version of the estimate $\widehat{f}_n$. Although on can use the recursive least squares algorithm to compute linear regression coefficients $\widehat{\theta}_{N,l}$ in (2.10), the derivations quickly become messy, because the number M of bins depends on N, and so does the number of equations in the algorithm.

Projection estimates

Another class of function estimates was introduced by Cencov [9], who called them *projection estimates.* The idea consists in expanding the unknown function into its "empirical" Fourier series. Consider the set $\mathcal{W}_2^s(L)$ of functions $f(x)$, $x \in [0,1]^d$, defined as follows. Each f can be represented by its Fourier series

$$f(x) = \sum_{|j|=1}^{\infty} c_j \Phi_j(x), \tag{2.11}$$

where $j = (j_1, \ldots, j_d)$ is a multi-index, $x = (x^1, \ldots, x^d)^T$, $\Phi_j(x) = \varphi_{j_1}(x^1) \times \ldots \times \varphi_{j_1}(x^d)$, $\varphi_1 \equiv 1$, $\varphi_{2k}(x) = \sqrt{2}\sin(2\pi k x)$ and $\varphi_{2k+1}(x) = \sqrt{2}\cos(2\pi k x)$, $k = 1, \ldots$. Suppose that the following condition is satisfied:

$$\sum_{j=1}^{\infty} |c_j|^2 (1 + |j|^{2s}) < L^2. \tag{2.12}$$

In fact, we have $\sum_{j=1}^{\infty} |c_j|^2 (1 + |j|^{2s}) \leq C\|f\|_{s,2}^2$, where $\|f\|_{s,2}$ is the norm of the Sobolev space $\mathcal{W}_2^s$ of functions with all derivatives up to order s being square integrable. Note that this is again a smoothness prior. We assume that *input X is uniformly distributed* [2]. We construct the estimate $\widehat{f}_N$ as follows:

$$\widehat{f}_N(x) = \sum_{j=1}^{m} \widehat{c}_j^N \Phi_j(x), \tag{2.13}$$

where m is the "model order", and the empirical estimates $\widehat{c}_j^N$ of Fourier coefficients

$$\widehat{c}_j^N = \frac{1}{N} \sum_{i=1}^{N} Y_i \Phi_j(X_i) \tag{2.14}$$

[2] See chapter 4 for a thorough discussion of this assumption.

are substituted for the true ones c_j, $j = 1, \ldots, m$. Note that the assumption that X is uniformly distributed has been used. Note also that the estimate (2.13)–(2.14) is linear (cf. (2.1)) with weights given by

$$W_{N,i}(x) = \sum_{j=1}^{m} \frac{1}{N} \Phi_j(x) \Phi_j(X_i).$$

Cencov, [9] has proved the following counterpart of statement 1 of theorem 1 : Let $\widehat{f}_N$ be a projection estimate. Then, uniformly over $f \in \mathcal{W}_2^s(L)$ and $x \in [0,1]^d$,

$$\mathbf{E}\|\widehat{f}_N(x) - f(x)\|_2^2 \leq C \left(L^2 m^{-2s} + \frac{\sigma_e^2 m^d}{N} \right). \tag{2.15}$$

The optimal order m of the model is

$$m = \lfloor \left(\frac{L^2 N}{\sigma_e^2} \right)^{1/(2s+d)} \rfloor , \tag{2.16}$$

it balances bias and variance error estimates, and yields the bound

$$\mathbf{E}\|\widehat{f}_N(x) - f(x)\|_2^2 \leq C L^{2/(d+2s)} \left(\frac{\sigma_e^2}{N} \right)^{2s/(2s+d)}. \tag{2.17}$$

The following result, due to Ibragimov and Khas'minskij [43], provides a global uniform bound. Take

$$m = \lfloor (\frac{N}{\ln N})^{1/(2s+d)} \rfloor$$

for the model order (note that this is slightly different from (2.16)). Then, uniformly over $f \in \mathcal{C}^s(L)$ (the class $\mathcal{C}^s(L)$ is defined in (2.5)), it holds that

$$\mathbf{E}\|\widehat{f}_N - f\|_\infty^2 \leq O\left(\frac{\ln N}{N} \right)^{2s/(2s+d)}. \tag{2.18}$$

Comments :

1. Projection estimates have the same rate of convergence (up to a constant) as kernel or piecewise polynomial ones.

2. The bound (2.15) for the quadratic error of the algorithms appears rather naturally if we consider the following argument: when we approximate $f \in \mathcal{W}_2^s$ using m terms of its Fourier decomposition, the approximation error is $O(m^{-2s/d})$. Furthermore, the stochastic error in each term is of order $O(N^{-1})$. This simple calculus can be repeated for any nonparametric estimate. Obviously, it is beyond our possibilities to reduce the stochastic component of the error. On the contrary, the bias part depends on the method we choose to approximate the function (piecewise polynomial, trigonometric series, etc.), and this choice of approximant is of primary importance.

3. From the computational point of view, projection estimates are more attractive than piecewise polynomial estimates, since it uses an orthonormal basis of functions (the Fourier basis), which dramatically simplifies the computation of the least-squares estimates $\widehat{c}_j$ of the Fourier coefficients c_j, cf. (2.14).

2.1.2 Practical implementation of the algorithms : adaptatation and tuning of their various design parameters, Generalized Cross Validation

As we have seen, the convergence of the estimates strongly depends on the choice of the bandwidth h_N for kernel estimator, the model order m for the projection estimator, and the binwidth δ_N (or, equivalently, the "model order" $M = \delta^{-1}$) for piecewise polynomial estimator. *These design parameters depend on the parameters of the smoothness class* $\mathcal{C}^s(L)$ *or* $\mathcal{W}_2^s(L)$*, which are a priori unknown* — see definition (2.5) of this class and the use of parameters (s, L) in Theorem 1 and corresponding results for others estimators. Even if some information about smoothness parameter s is available, the knowledge of the value L is of importance when the data sample is of bounded length. Let us illustrate this with the following example, where input x is scalar. Consider the problem of estimating a function $f(x)$ in additive white noise e, with $\sigma_e^2 = 1$. Assume that f has support $[0, 1]$, that all its derivatives are continuous, and that $f(1/2) = 1$, $f(0) = f(1) = 0$. Note that in this case, typically, $\sup_x |f^{(s)}(x)| \approx s^s$, i.e., higher order derivatives become very large in uniform bound. In this case the bounds in Theorem 1 are of order $a_N(s) = (s/N)^{2s/(2s+1)}$ when the parameter is selected for the smoothness s. Assume that the size of the observation sample is $N = 10000$, then $a_N(2) = 0.0110$, $a_N(3) = 0.0095$, but we have already $a_N(4) = 0.0122$ (the value of s which minimizes a_N is $s \approx 3.4814$ with $a_N(s) \approx 0.00946$). This illustrates the fact that the tightest bound is not obtained by taking the largest possible s, but rather by selecting the most favourable pair (s, L), which is obviously much more difficult.

Given that we only have in practice samples of finite size N, we shall not try to estimate the most favourable pair (s, L), but we shall proceed differently. The model order (or bandwith, or binwidth, depending on the different estimates) shall be estimated from data using a procedure usually referred to as the *Generalized Cross Validation* (GCV) test. GCV procedures were studied for kernel (see, for instance, [69], [40]), spline ([49], [12]), and projection estimates (c.f. [66], [50]). Let us consider, for instance, the procedure for the projection estimates [3]. To make the model order explicit in formula (2.13) we shall write $\widehat{f}_{m,N}$ instead of $\widehat{f}_N$. Set $S_{m,N}^2 = N^{-1} \sum_{i=1}^N \|Y_i - \widehat{f}_{m,N}(X_i)\|^2$. As for the prediction error variance estimate in parametric prediction error methods, $S_{m,N}^2$ is a *biased* estimate of the error. Thus one cannot minimize $S_{m,N}^2$ with respect to m directly : the result of such a bruteforce procedure would give a function $\widehat{f}_{m_N,N}(x)$ which perfectly fits the noisy data, this is known as "overfitting" in the neural network litterature. The solution rather consists in introducing

[3] In fact, a similar result holds for the spline or piecewise polynomial ones.

a penalty which is proportional to the model order m, i.e., we search for m_N such that

$$m_N = \arg \min_{m \leq N} \left(S^2_{m,N} + \frac{2\sigma_e^2 m}{N} \right). \tag{2.19}$$

This technique is clearly equivalent to the celebrated Mallows-Akaike criterion [54], [1]. The following result, due to Polyak and Tsybakov [66], shows the consistency of this procedure. Assume that the Fourier coefficients of f in expansion (2.11) satisfy $|c_j| \leq \varepsilon_j$, $\sum_{j=1}^{\infty} \varepsilon_j < \infty$, $(j\, \varepsilon_j)$ is non-increasing, and σ_e^2 is known. Set $V_{m,N} = \|\widehat{f}_{m,N} - f\|_2^2$. Then for the estimate (2.13), (2.14), and (2.19), it holds that

$$\frac{V_{m_N,N}}{\min_m V_{m,N}} \to 1 \text{ a.e. as } N \to \infty .$$

2.2 Performance analysis of the nonparametric estimators

The performance analysis of nonparametric estimation algorithms and/or identification procedures is much more difficult than for parametric estimation. In fact, the following specific issues are important:

1. What plays the role of Cramer-Rao bound and Fisher Information Matrix in our case? Recall that the Cramer-Rao bound reveals the best performance one can expect in identifying the unknown parameter θ from sample data arising from some parametrized distribution $\mathbf{p}_\theta, \theta \in \Theta$, where Θ is the domain over which the unknown parameter θ ranges. In the nonparametric case, lower bounds for the best achievable performance are provided by *minimax risk functions.* We shall introduce these lower bounds and discuss associated notions of optimality.

2. For lower bounds, what is the class of systems on which best achievable performance is considered, is another important issue. For nonparametric representations of linear systems, $L_2, L_\infty, H_2, H_\infty$, with their associated norm are typical spaces to work with. For (even static) nonlinear systems, however, the choice is much wider. How wide should be the class $\mathcal{F}$ of systems in consideration, what kind of smoothness should be required? Are we interested in the behaviour of the estimate at one particular point x of interest, or are we interested in the global behaviour of the estimate? Different distance measures should be used in these two different cases.

2.2.1 Lower bounds for best achievable performance

In order to compare different nonparametric estimators it is necessary to introduce suitable figures of merit. It seems first reasonable to build on the mean square deviation (or mean absolute deviation) of some semi-norm [4] of the error, we denote it by $\|\widehat{f}_N - f\|$. The following

[4] a seminorm is a norm, except it does not satisfy the condition: $\|f\| = 0$ implies $f = 0$.

semi-norms are commonly used in nonparameteric regression: $\|f\| = \left(\int f^p(x)dx\right)^{1/p}$, $0 < p < \infty$ (L_p-norm), $\|f\| = \sup_x |f(x)|$ (uniform norm, $\mathcal{C}$- or L_∞-norm), $\|f\| = |f(x_0)|$ (absolute value at a fixed point x_0). Then we consider the *risk function*

$$R_{a_N}(\widehat{f}_N, f) = \mathbf{E}\left[a_N^{-1}\|\widehat{f}_N - f\|\right]^2, \tag{2.20}$$

where a_N is a normalizing positive sequence. Letting a_N decrease as fast as possible so that the risk still remains bounded yields a notion of a convergence rate. Let $\mathcal{F}$ be a set of functions which contains the "true" regression function f, then the maximal risk $r_{a_N}(\widehat{f}_N)$ of estimator $\widehat{f}_N$ on $\mathcal{F}$ is defined as follows:

$$r_{a_N}(\widehat{f}_N) = \sup_{f\in\mathcal{F}} R_{a_N}(\widehat{f}_N, f) \ .$$

If the maximal risk is used as a figure of merit, the optimal estimator $\widehat{f}_N^*$ is the one for which the maximal risk is minimized, i.e., such that [5]

$$r_{a_N}(\widehat{f}_N^*) = \min_{\widehat{f}_N} \sup_{f\in\mathcal{F}} R_{a_N}(\widehat{f}_N, f) \ .$$

We call $\widehat{f}_N^*$ *the minimax estimator* and the value

$$\min_{\widehat{f}_N} \sup_{f\in\mathcal{F}} R_{a_N}(\widehat{f}_N, f)$$

the minimax risk on $\mathcal{F}$. The construction of minimax nonparametric regression estimators for different sets $\mathcal{F}$ is a hard problem. Today, it is only solved asymptotically (for large samples) for some special cases (see, for instance, [34], [35], [36]). However, letting a_N decrease as fast as possible so that the minimax risk still remains bounded yields a notion of a best achievable convergence rate, similar to that of parametric estimation. More precisely, we state the following definition:

Definition 1 (lower rate and minimax rate of convergence)

1. *The positive sequence a_N is a* **lower rate of convergence for the set $\mathcal{F}$ in the semi-norm $\|\cdot\|$** *if*

$$\liminf_{N\to\infty} r_{a_N}(\widehat{f}_N^*) = \liminf_{N\to\infty} \inf_{\widehat{f}_N} \sup_{f\in\mathcal{F}} \mathbf{E}\left[a_N^{-1}\|\widehat{f}_N - f\|\right] \ \geq \ C_0 \tag{2.21}$$

for some positive C_0. The inequality (2.21) is a kind of negative statement that says that no estimator of function f can converge to f faster than a_N. This notion can be refined as follows.

[5] to properly understand the statement to follow, the reader should pay attention to definition (1.3) of an estimator.

2. *The positive sequence a_N is called* **minimax rate of convergence** *for the set $\mathcal{F}$ in semi-norm $\|\cdot\|$, if it is lower rate of convergence, and if, in addition, there exists an estimator $\widehat{f}_N^*$ achieving this rate, i.e., such that*

$$\limsup_{N\to\infty} r_{a_N}(\widehat{f}_N^*) \;<\; \infty \;.$$

Thus, a coarser, but easier approach consists in assessing the estimators by their convergence rates. In this setting, by definition, optimal estimators reach the lower bound as defined in (2.21) (recall that the minimax rate is not unique : it is defined to within a constant).

Some negative results. We state first a negative result, due to Devroye and Györfi [21] [20], which expresses that no convergence rate exists if no smoothness assumption about the unknown regression function f is stated [6]. Consider the following classes of functions on $\mathbf{R}$:

$\mathcal{F}^*$: the class of all functions f such that $f(x) = 0$ for $x > 1$ or $x < 0$, and $|f(x)| \le C$ for $x \in [0,1]$.

$\mathcal{F}_0^*$: the class of all continuous functions $f \in \mathcal{F}^*$.

$\mathcal{F}_\infty^*$: the class of all functions $f \in \mathcal{F}^*$ having all continuous derivatives on $[0,1)$ (be careful that the interval is right open).

Let $\widehat{f}_N$ be an arbitrary estimate of f. Then for the classes $\mathcal{F}^*$, $\mathcal{F}_0^*$ and $\mathcal{F}_\infty^*$ defined above (we denote them generically by $\mathcal{F}$),

$$\sup_{\mathcal{F}} \limsup_{N\to\infty} \mathbf{E}\left[a_N^{-1}\int_0^1 |\widehat{f}_N(x) - f(x)|dx\right] = \infty$$

for any positive sequence $a_N \to 0$.

There is also a similar result for the adaptive classification problem : consider the classification problem of section 1.2 and notations therein. Suppose that there are only two classes, i.e., $M = |\mathcal{Z}| = 2$. Let a_N be any positive sequence such that $a_N \to 0$, and $\lambda \in [0, 1/2)$. Let g_N be an arbitrary estimator. Then there exists a distribution of the pair (X, Z), with X uniformly distributed on $[0,1]$, such that

$$\limsup_{n\to\infty} a_N^{-1}(\mathbf{E}\mathcal{L}_N - \mathcal{L}^*) = \infty \;,$$

where $\mathcal{L}_N$ is associated with g_N through (1.7).

Thus, *no convergence rate does exist for any of the above classes $\mathcal{F}^*$, $\mathcal{F}_0^*$ and $\mathcal{F}_\infty^*$*. In other words, the convergence can be arbitrary slow, depending on the unknown function or density f to be estimated ! It is a natural consequence of the fact that the above classes $\mathcal{F}^*$, $\mathcal{F}_0^*$ and $\mathcal{F}_\infty^*$ are too rich : they contain functions which are extremely difficult to approximate. *In other words, in order to obtain any interesting rate of convergence, smoothness conditions should be imposed.*

[6] Note that convergence can sometimes be proved without any smoothness assumption [22].

Some positive results. Let us now concentrate on the case of deterministic uniform design, i.e., the input data X are uniformly sampled in the considered interval. The following result in the case of regular design can be acknowledged to [43] (for the random design case, see [74], [47]).

Theorem 2 *Let us consider the Hölder class $\mathcal{C}^s(L)$ on $[0,1]^d$, see (2.5) for the definition of $\mathcal{C}^s(L)$. Consider*

$$\|g\| = \left(\int |g(x)|^p dx \right)^{1/p}, \quad 0 < p < \infty \qquad or \quad \|g\| = |g(x_0)| \ .$$

Then $N^{-\frac{s}{2s+d}}$ is a lower rate of convergence for the class $\mathcal{C}^s(L)$ in the semi-norm $\|\cdot\|$. Furthermore, $\frac{N}{\ln N}^{-\frac{s}{2s+d}}$ is a lower rate of convergence for the class $\mathcal{C}^s(L)$ in the norm $\|g\| = \sup_{x\in[0,1]} |g(x)|$.

Note that to obtain the correct rate of convergence for the distance at a fixed point x_0, the corresponding Lipschitz property is required at x_0 only. Similar results hold when the class $\mathcal{C}^s(L)$ is replaced by the class $\mathcal{W}_p^s(L)$, $p \geq 2$, where $\mathcal{W}_p^s(L)$ is the set of k-times differentiable functions f on $[0,1]^d$ such that $\|f\|_2 \leq 1$, $\|f^{(k)}(t+h) - f^{(k)}(t)\|_p \leq L\|h\|^\alpha$, $0 < \alpha \leq 1, s = k + \alpha$ [7]. Then $N^{-\frac{s}{2s+d}}$ is also a lower rate of convergence for this class in L_p-norm of the error.

2.2.2 Discussion

Criticizing the minimax paradigm. Let us compare the lower rates of convergence of theorem 2 and the upper bounds obtained in this section for different estimators. One can see that the estimators considered are optimal on the classes $\mathcal{W}_2^s$ and $\mathcal{C}^s$ in the sense that they reach the minimax optimal rate of convergence [8]. Despite many impressive technical achievements in the above work, the general reaction within the statistics community has not been really enthusiastic. For example, David Donoho quotes that *"... a large number of computer packages appeared over last fifteen years, but the work on the minimax paradigm has relatively little impact on software"* [29]. One of the arguments supporting this skepticism about methods based on the minimax paradigm — kernel estimators, spline methods or orthogonal series — is that they are spatially nonadaptive, while real functions exhibit a variety of shapes and spatial inhomogeneities. To illustrate this point let us look at the following example. Consider the function $f(x) = 1_{\{0 \leq x < a\}}$ for some $0 < a < 1$. The Fourier coefficients of this function are

$$c_0 = a, \quad c_{2k} = \sqrt{2}\, \frac{\sin^2(\pi k a)}{\pi k}, \quad c_{2k+1} = \sqrt{2}\, \frac{\sin(\pi k a)\cos(\pi k a)}{\pi k},$$

[7] Although defined in a different way, this $\mathcal{W}_p^s(L)$ space coincides for $p = 2$ with the space introduced in formula (2.11) and subsequent ones.

[8] the projection estimates are also minimax on $\mathcal{W}_p^s$ (see th. 4.3 [43])

hence the condition in (2.12) is not verified for $s \geq 1/2$. Thus, we conclude from (2.18) that the rate of convergence (2.17) for the projection estimate (2.13), (2.14) will not be better than $N^{-1/2}$. Furthermore, since f does not belong to the Sobolev space $\mathcal{W}_2^s$ for $s \geq 1/2$, this rate of convergence is minimax. On the other hand, one naturally expect that a procedure to detect the edges of f can be designed which would have a rate of convergence "close" to N^{-1}. Indeed, the linear methods fit very well functions which are, say, "uniformly smooth" or "uniformly non-smooth". Facing the problem of estimating a function with sparse singularities, the projection method will infer erroneously that the function is "uniformly smooth", but with a pessimistic smoothness parameter.

The minimax paradigm as discussed before does not seem to provide methods with convergence rates of order N^{-1} for the above example, thus the authors of [29] argue that one should construct methods (heuristically, if necessary) which address the "real problem", namely *spatial adaptation.* This point of view has had considerable influence on software development and daily statistical practice, apparently much more than the minimax paradigm. Interesting spatially adaptive methods include all sorts of neural networks, projection pursuit [38], classification and regression trees (CART) [8], Multivariate adaptive regression splines (MARS) [37], Variable Bandwidth Kernel methods [57], and others. These methods implicitly or explicitly attempt to adapt the fitting method to the form of the function being estimated, by ideas like recursive dyadic partitioning of the space on which the function is defined (CART and MARS) and adaptively estimating a local bandwidth function (Variable Kernel Methods). Citing again David Donoho, one could say that *"the spatial adaptivity camp is, to date, a-theoretical, as opposed to anti-theoretical, motivated by the heuristic plausibility of their methods, and pursuing practical improvements rather than hard theoretical results which might demonstrate specific quantitative advantages of such methods. But, in our experience, the need to adapt spatially is so compelling that the methods have spread far in the last decade, even though the case for such methods is not proven rigorously"* [29]. To conclude, a deeper investigation is needed to find the proper framework.

The adequate answer: Besov spaces and wavelets. This short analysis reveals the crux in the route to both practical efficiency and mathematical support of the methods. It consists in *finding a parametrized family of functional classes which*

1. *fits our prior knowledge about the smoothness of the function to be estimated, (in particular, that f is smooth everywhere, except at a sparse set of points), and*
2. *has associated with it an estimation technique which is minimax within these classes.*

It was the merit of David Donoho and Iain Johnstone [25] to recognize that Besov spaces, which play a central role in Yves Meyer's mathematical theory of wavelets [55], provide an adequate answer. They are perfectly suited to nonlinear systems which have sparse singularities and otherwise are smooth. This material will be the topic of Chapter 4.

However, before discussing wavelets and their use in identification, we briefly scan some popular nonlinear estimates. They all provide the kind of "spatial adaptation" that we advocated before. Some of them are supported by efficient software. And some of them have

become extremely succesfull and their name are now buzzwords widely known beyond the scientific community.

2.3 Nonlinear estimates

Starting from early 1980's a variety of techniques have been proposed in the statistics litterature, which exhibit this desirable feature of "spatial adaptivity". Among them *Projection Pursuit Algorithm* developed in [38] (very good review of these results can be found in [41]), *Recursive Partitioning* [56], [8] and related methods (c.f., for instance [37] with discussion). These methods are derived from some mixture of statistic and heuristic arguments and give impressive results in simulations. Their drawback lies in the almost total absence of any theoretical results on their convergence. We refer the reader to the above references for additional information.

Surprinsingly enough, the A.I. litterature has proposed independently and at the same time different techniques with the same feature of "spatial adaptivity". These include various forms of neural networks [42], see the other tutorial [52] by Lennart Ljung. We shall briefly describe these. In addition we shall sketch a recent technique due to Leo Breiman [7], which practically combines some advantages of neural networks (in particular the ability to handle very large dimensional inputs) and of constructive wavelet based estimators (availability of very fast training algorithms).

A relationship with neural networks; A. Barron's result

The following result was recently published in [2], it is the most accurate theoretical result about neural networks available today. Let $\sigma(x)$ be a sigmoidal function (i.e. a bounded measurable function on the real line for which $\sigma(x) \to 1$ as $x \to \infty$ and $\sigma(x) \to 0$ as $x \to -\infty$). Consider a compactly supported function f with $\text{supp}\ (f) \subseteq [0,1]^d$, and assume that

$$C_f = \int_{\mathbf{R}^d} |\omega|\ |\widehat{f}(\omega)|\ d\omega < \infty\ , \tag{2.22}$$

where $\widehat{f}(\omega)$ denotes the Fourier transform of f. The main result of [2] can be roughly stated as follows: there exists an approximation f_n of the compactly supported function f, of the form

$$f_n(x) = \sum_{i=1}^{n} c_i \sigma(a_i^T x + t_i) + c_0 \tag{2.23}$$

(note that f_n is *not* compactly supported), such that

$$\|(f_n - f)\ 1_{[0,1]^d}\|_2\ \leq\ 2\sqrt{d}\ C_f\ n^{-1/2}\ . \tag{2.24}$$

This result provides an upper bound of the minimum distance (in L_2-norm) between any f satisfying condition (2.22) and the class of all neural networks of size not larger than n. In the

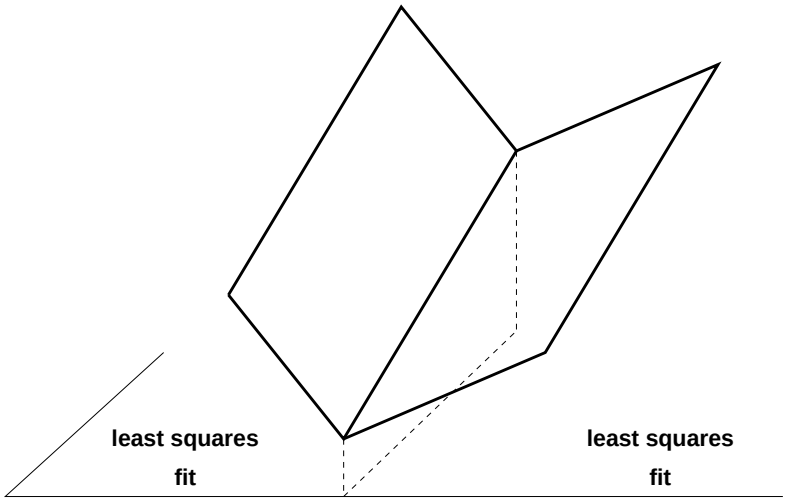

Figure 2.1: *A hinge function on* $\mathbf{R}^2$*. On each side of the corner, best fit is just performed via linear least squares.*

same article, this upper bound is compared with the best achievable convergence rate for any linear estimator in class (2.22). It is shown that a lower rate for linear estimators is $n^{-1/d}$, compare with the much better rate $n^{-1/2}$ for neural networks, especially for large dimension d. No result is available which takes advantage of possible improved smoothness of the unknown system f. An iterative algorithm for the construction of the approximation (2.23) is also proposed. The true problem of system identification, i.e., that of neural network training based on noisy input/output data, is not addressed in this paper. Also, neural networks need the backpropagation procedure for their training, a stochastic gradient procedure which is known to be of prohibitive cost. In turn, neural network training works even for very large dimensional input data.

Breiman's Hinging Hyperplanes

We now briefly discuss a recent technique due to Leo Breiman [7], which practically combines some advantages of neural networks (in particular the ability to handle very large dimensional inputs) and of constructive wavelet based estimators (availability of very fast training algorithms). Breiman's technique is a very elegant and efficient way of identifying piecewise linear models based on data collected from an unknown nonlinear system, see [73] for the use of such models in control. Following [7], we call *hinge function* a function $y = h(x)$, $x \in \mathbf{R}^d$ which consists of two hyperplanes continuously joined together, i.e., an open book, see figure 2.1. If the two hyperplanes are given as

$$y = \langle \beta^+, x \rangle + \beta_0^+, \quad y = \langle \beta^-, x \rangle + \beta_0^-,$$

where $\langle ., . \rangle$ denotes scalar product in Euclidian spaces, then an explicit form for the hinge function is either

$$\begin{aligned} & h(x) = \max(\langle \beta^+, x \rangle + \beta_0^+ , \langle \beta^-, x \rangle + \beta_0^-) , \\ \text{or} \quad & h(x) = \min(\langle \beta^+, x \rangle + \beta_0^+ , \langle \beta^-, x \rangle + \beta_0^-) . \end{aligned}$$

It is proved in [7], using the methods by Barron [2] that there is a constant C such that for any n there are hinge functions $h_1, \ldots, h_n$ such that

$$\| f - \sum_{i=1}^{n} h_i 1_{[0,1]^d} \|_2 \leq C n^{-1/2} \tag{2.25}$$

for any f such that

$$\int_{\mathbf{R}^d} |\omega|^2 |\widehat{f}(\omega)| d\omega < \infty ,$$

i.e., Breiman's hinge model is as efficient as neural networks for the L_2-norm. An iterative projection algorithm is proposed to compute the approximation. The interesting point about this iterative approximation technique is that it converges with a magnitude order faster than backpropagation does. To understand why this can happen, consider the simplest case where x is of dimension 1, f itself is a hinge function, and we try to fit a single hinge approximant (i.e., $n = 1$ in (2.25)). Thus we have to estimate the four unknown parameters $(\beta^{\pm}, \beta_0^{\pm})$. This is done iteratively as follows. First, guess the corner of the hinge (i.e., the x where both arguments in the "max" or "min" are equal), call it $x(0)$. Selecting only those $x > x(0)$ with corresponding y's, a first estimate for, say, (β^+, β_0^+) is obtained by ordinary linear least squares fit, and similarly for $x < x(0)$. Thus we now have a first hinge $\widehat{f}(1)$, which yields a new corner $x(1)$, and so on. This converges extremely rapidly. In contrast there is no such fast procedure for a single neuron with adjustable parameters to estimate an unknown single neuron, since stochastic gradient must be used even in this case. A method based on nested iterations of the above kind is proposed in (2.25) to fit general f's. Reported experimental results show the efficiency of this technique. These experiments show that practically the approximation obtained is much more accurate than it is suggested by the estimate in (2.25). On the other hand, note that a superposition of hinge functions is not smooth, since it is piecewise linear. Also the use of superposition of hinge functions is especially advocated in (2.25) for large dimensional x's. However, as indicated at the beginning of this section, no convergence rate is given for models identified from noisy data (the bound (2.25) is not a convergence rate for identification, but only a rate of approximation of a given function by some finitely parametrized class of approximants).

Chapter 3

Wavelets : what they are, and their use in approximating functions

WARNING : throughout this chapter, the notation $\widehat{\varphi}(\omega)$ denotes the Fourier transform of function $\varphi(x)$, and *not* the estimator of φ.

3.1 The continuous wavelet transform

The continuous wavelet transform and inverse transform of a function f are respectively given by equations (3.2) and (3.3) below. These transforms use two functions $\psi(x)$ and $\varphi(x) \in L_2(\mathbf{R}^d)$, both radial (i.e., depending only on $|x|$), known as the *analysis and synthesis wavelets*:

Theorem 3 *Let ψ and φ be radial functions satisfying*

$$\forall \omega \in \mathbf{R}^d \ : \ \int_0^\infty a^{-1} \widehat{\varphi}(a\omega)\widehat{\psi}(a\omega) da = 1 \tag{3.1}$$

where we recall that $\widehat{\varphi}(\omega)$ denotes the Fourier transform of function $\varphi(x)$. Then for any function $f \in L_2(\mathbf{R}^d)$, the following formulae define an isometry between $L_2(\mathbf{R}^d)$ and a subspace of $L_2(\mathbf{R}^d \times \mathbf{R}_+)$ [14]:

$$u(a,t) \ = \ a^{d-1/2} \int f(x) \ \psi(a(x-t)) \ dx \tag{3.2}$$

$$f(x) \ = \ \int u(a,t) \ \varphi(a(x-t)) \ a^{d-1/2} \ da \ dt \ . \tag{3.3}$$

Here $a \in \mathbf{R}^+$ and $t \in \mathbf{R}^d$ are respectively the dilation and translation factors. Note that the integral (3.1) does not depend on $\omega \neq 0$ since the functions ψ and φ are radial. In order for this integral to be properly defined, it is sufficient that, for example $\widehat{\varphi}(\omega)\widehat{\psi}(\omega) = O(|\omega|)$; this happens if $\varphi(x)$ and $(1+|x|)\psi(x)$ are in $L_1(\mathbf{R}^d) \cap L_2(\mathbf{R}^d)$ and ψ has zero integral. Once the integral (3.1) is well defined and finite, a simple normalization leads to a pair (φ, ψ) which satisfies the assumption.

Examples: One can verify that the following pairs ψ, φ satisfy the assumption :

$$\psi(x) = \sqrt{2}(d - |x|^2)e^{-\frac{|x|^2}{2}} \ , \quad \varphi(x) = \sqrt{2}e^{-\frac{|x|^2}{2}}$$
$$\psi(x) = \varphi(x) = \frac{1}{\sqrt{2}}(d - |x|^2)e^{-\frac{|x|^2}{2}}$$

and, in the one dimensional case :

$$\psi(x) = -\text{sign}(x)\ 1_{\{|x|<1\}} \ , \quad \varphi(x) = \frac{1-|x|}{3}\ 1_{\{|x|<1\}}$$
$$\psi(x) = -1_{\{-1\leq x<-\frac{1}{2}\}} + 1_{\{-\frac{1}{2}\leq x<\frac{1}{2}\}} - 1_{\{\frac{1}{2}\leq x<1\}} \ , \quad \varphi(x) = \lambda^{-1}e^{-\frac{x^2}{2}}$$

with $\lambda = -0.03527343656\ldots$ and $1_{\{A\}}$ is the indicator function of the set A. The choice of possible pairs ψ, φ is very large. In particular, pairs (ψ, φ), with ψ nonsmooth but φ smooth, are allowed.

Time-frequency localization: even this simple construction provides a very interesting property: roughly speaking, the behavior of function $u(a,t)$, when scaling factor a is fixed, measures the smoothness of f in the neighbourhood of point t. This focusing effect is called "time-frequency localization" (see discussion in chapter 2 of [14]). It is not provided by the Fourier transform (the behavior of the Fourier transform $\widehat{f}(\omega)$ reflects the *global* smoothness of f). Unfortunately, these localization properties of continuous wavelet transform cannot be used for estimation, because there is no associated algorithm to compute this transform. For practical purposes the reconstruction formula (3.3) has to be discretized :

$$f(x) = \sum_i u_i\ \varphi(a_i x - t_i) \ , \tag{3.4}$$

this point will be discussed in Section 3.2 and in the next chapters.

3.2 The discrete wavelet transform : orthonormal bases of wavelets and extensions

Multiresolution analysis introduced by Stephane Mallat and further developed by Ingrid Daubechies provides orthonormal bases of $L_2(\mathbf{R})$ of the form $\psi_{j,k}(x) = \{2^{j/2}\psi(2^j x - k) \ : \ j,k \in \mathbf{Z}\}$, i.e., each element of the basis is a translated and dilated version of a

single *wavelet* ψ. For a function $f \in L_2(\mathbf{R})$, the inner product $\langle f, \psi_{j,k} \rangle$ performs zooming on f over a $O(2^{-j})$ width interval centered at point $2^{-j}k$. Thus, *large j corresponds to checking function f at fine scales.* This implies that a local singularity of a function f will affect only a small part of its coefficients in this wavelet basis. This is the main difference with the Fourier basis : a local singularity of f would affect the whole Fourier representation.

3.2.1 Definition and construction of orthogonal wavelet bases

To begin we first discuss the scalar case, i.e., that of functions defined on $\mathbf{R}$. Otherwise explicitly stated, all results in this subsection are borrowed from monograph [14].

Definition 2 (Multiresolution Analysis (MA)) *A multiresolution analysis consists of a function φ, $\|\varphi\|_2 = 1$, and a sequence $(V_j)_{j \in \mathbf{Z}}$ of spaces defined by*

$$\begin{aligned} \varphi_{jk} &= 2^{j/2}\varphi(2^j x - k) \quad , \; j,k \in \mathbf{Z} \\ V_j &= \text{Span } \{\varphi_{jk}, k \in \mathbf{Z}\} \end{aligned}$$

with the properties:

(MA0): $(\varphi_{0k})_{k \in \mathbf{Z}}$ *is an orthonormal family*

(MA1): $\bigcap_{j \in \mathbf{Z}} V_j = \{0\}$

(MA2): $\overline{\bigcup_{j \in \mathbf{Z}} V_j} = L_2(\mathbf{R})$

(MA3): $V_j \subset V_{j+1}$

Property (MA3) is equivalent to the existence of a square integrable sequence (h_k) such that

$$\varphi(x) = \sqrt{2} \sum h_k \; \varphi(2x - k) \; . \tag{3.5}$$

We call such a function the *scale function* (also known as the *father wavelet* [55]). Theorem 4 to follow is the basis of the theory ; it shows how, starting from a multiresolution analysis and its scale function φ, we can construct very simply an orthonormal basis of $L_2(\mathbf{R})$.

Theorem 4 *Assume that conditions (MA0–3) are satisfied. Set* [1]

$$\begin{aligned} \psi(x) &= \sqrt{2} \sum g_k \; \varphi(2x - k) \; , \quad g_k = (-1)^{k+1} \; \overline{h}_{1-k} \\ \psi_{jk} &= 2^{j/2} \; \psi(2^j x - k) \\ W_j &= \text{Span}(\psi_{jk}, k \in \mathbf{Z}) \end{aligned} \tag{3.6}$$

then

1. $V_{j+1} = V_j \bigoplus W_j$ *and* $\{\psi_{jk} \; : \; j,k \in \mathbf{Z}\}$ *is an orthonormal basis in* $L_2(\mathbf{R})$ *;*

[1] $\overline{h}$ denotes the complex conjugate of h.

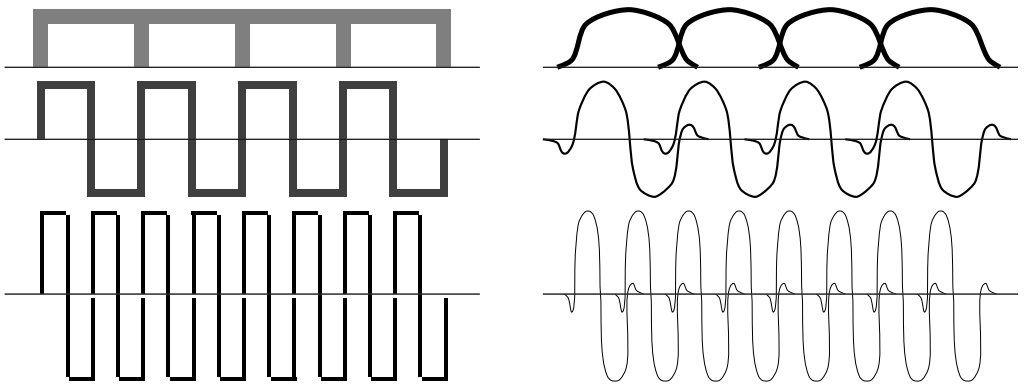

Figure 3.1: *The Haar basis (left side) and a wavelet basis (right side). The first row shows the scale function φ and the subsequent rows show wavelets ψ at two successive scales.*

2. $L_2(\mathbf{R}) = V_0 \bigoplus W_0 \bigoplus W_1 \bigoplus \ldots$ *and* $\{\varphi_{0k}, \psi_{jk} \ : \ j \geq 0, k \in \mathbf{Z}\}$ *is an orthonormal basis in* $L_2(\mathbf{R})$.

The function $\psi(x)$ defined in (3.6) is often referred to as the "mother wavelet".

Multiresolution analysis and orthonormal wavelets are pictured in Figure 3.1. Then theorem 5 gives the basic tool for building scale functions.

Theorem 5 *Let $m_0(\omega)$ be a trigonometric polynomial*

$$m_0(\omega) = \frac{1}{\sqrt{2}} \sum_{k=K}^{L} h_k e^{-ik\omega}$$

such that

(QMF1): $m_0(0) = 1$,

(QMF2): $m_0(\omega) \neq 0$ *if* $\omega \in [-\pi/2, \pi/2]$,

(QMF3): $|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1$.

Then the function φ, with Fourier transform given by

$$\widehat{\varphi}(\omega) = \prod_{j=1}^{\infty} m_0(2^{-j}\omega)$$

satisfies assumptions (MA0-3) and supp$(\varphi) \subset [K, L]$.

Examples of polynomials satisfying assumption (QMF1–3) are given in [14] and smoothness properties of φ and ψ are studied. Links with multirate digital signal processing and Quadrature Mirror Filter (QMF) banks are discussed in [4], see the next subsection.

We now move on discussing the multidimensional case. There exist two main types of constructions of the wavelet basis with dilation factor 2 in $\mathbf{R}^d$ ([14], 10.1). A first guess simply consists in taking tensor product functions generated by d one-dimensional bases :

$$\Psi_{j_1,k_1,\ldots,j_d,k_d}(x) = \psi_{j_1,k_1}(x_1) \times \ldots \times \psi_{j_d,k_d}(x_d). \tag{3.7}$$

This construction has the drawback of mixing different resolution levels j_i. Alternatively, if such a mixing is not desired, we proceed as follows. Introduce the scale function

$$\Phi(x) \quad = \quad \varphi(x_1) \times \ldots \times \varphi(x_d) \tag{3.8}$$

and the $2^d - 1$ mother wavelets $\Psi^{(l)}(x)\,, i = 1, \ldots, 2^d - 1$ obtained by substituting in (3.8) some $\varphi(x_j)$'s by $\psi(x_j)$'s. Then the following family is an orthonormal basis of $L_2(\mathbf{R}^d)$:

$$\left\{\Phi_{0k}(x), \Psi_{jk}^{(1)}(x), \ldots, \Psi_{jk}^{(2^d-1)}(x)\right\};\ j \in \mathbf{N}_0,\ k = (k_1, \ldots, k_d) \in \mathbf{Z}^d \tag{3.9}$$

where $\mathbf{N}_0 = \mathbf{N} \cup 0$, and

$$\begin{aligned} \Phi_{jk}(x) &= 2^{jd/2}\ \Phi(2^j x_1 - k_1, \ldots, 2^j x_d - k_d) \\ \Psi_{jk}^{(l)}(x) &= 2^{jd/2}\ \Psi^{(l)}(2^j x_1 - k_1, \ldots, 2^j x_d - k_d)\ . \end{aligned}$$

NOTA : as formula (3.9) shows, constructing and storing orthonormal wavelet bases become of prohibitive cost for large dimension d. This is the main limitation for using the otherwise very efficient techniques which rely on orthonormal wavelet bases (and their generalizations).

3.2.2 Orthogonal wavelet bases and Quadrature Mirror Filters (QMF)

For the sake of simplicity, we only discuss the one dimensional case. Equations (3.5) and (3.6) imply that [2], for $f \in L_2(\mathbf{R})$,

$$\alpha_{jk} \quad = \quad \langle f, \varphi_{jk} \rangle, \quad \beta_{jk} = \langle f, \psi_{jk} \rangle \tag{3.10}$$

satisfy [3]

$$\alpha_{jk} \quad = \quad \sum_l \overline{h}_{l-2k}\ \alpha_{j+1,l} \tag{3.11}$$

$$\beta_{jk} \quad = \quad \sum_l \overline{g}_{l-2k}\ \alpha_{j+1,l}. \tag{3.12}$$

[2] recall that $\langle .,. \rangle$ denotes the inner product in L_2.
[3] recall that $\overline{h}$ denotes the complex conjugate of h.

Introduce the polynomial filters

$$H(z) = \sum_k h_k z^{-k} \quad , \quad G(z) = \sum_k g_k z^{-k} \tag{3.13}$$

where coefficients h_k, g_k are as in (3.5) (3.6). Also denote by $\downarrow^{(2)}$ the decimation of a signal by a factor of two:

$$\downarrow^{(2)} (x_n) \quad = \quad (x_{2n})$$

Thus, if we consider α_{jk} as a signal indexed by k and denote it by α_j, relations (3.12) translate into

$$\alpha_j = \downarrow^{(2)} \; H \; \alpha_{j+1} \quad , \quad \beta_j = \downarrow^{(2)} \; G \; \alpha_{j+1}$$

and property (QMF3) expresses that the pair (H, G) is QMF [78] [4]. Equations (3.11) and (3.12) are used to compute recursively from fine scales to coarse scales the orthonormal wavelet decomposition. Assume that, in addition, scale function φ is selected so that the computation of inner product $\langle f, \varphi_{jk} \rangle$ in (3.10) is performed efficiently for some scale j. Then, *formulas (3.10), (3.11), and (3.12) together build a highly efficient procedure for computing the wavelet decomposition of f.* As pointed out at the end of the preceding subsection, orthonormal wavelet bases become of prohibitive storage cost for large dimension d, however. Scale functions φ are proposed in [14], with vanishing moment conditions, for which

$$\langle f, \varphi_{jk} \rangle \quad = \quad f(2^{-j} k) + O(2^{-Mj}) \tag{3.14}$$

holds, where integer M is related to the number of vanishing moments (such scale functions are often referred to as "coiflets"). Note that the above approximation is at the same time good and very easy to compute. Alternative techniques to get simple approximations similar to (3.14) are proposed in [23] and [16].

Since QMF pairs are known to allow exact reconstruction of filtered-and-decimated signals [78] [4], equations (3.11) and (3.12) can be "inverted" to yield the synthesis equation

$$\alpha_{jk} \quad = \quad \sum_l h_{k-2l} \; \alpha_{j-1,l} + g_{k-2l} \; \beta_{j-1,l}. \tag{3.15}$$

For $f \in V_{j_0}$, we have, by definition of this space,

$$f = \sum_k \alpha_{j_0 k} \; \varphi_{j_0 k} \; , \tag{3.16}$$

and, since $V_{j_0} = V_0 \bigoplus W_0 \bigoplus W_1 .... \bigoplus W_{j_0}$,

$$f = \sum_k \alpha_{0k} \; \varphi_{0k} + \sum_{j,k} \beta_{jk} \; \psi_{jk} \; . \tag{3.17}$$

Formulas (3.11) and (3.12) allow us to switch from representation (3.16) to representation (3.17). The latter one is generally much more compact since, when f is smooth, most β_{jk} are

negligible. In the multidimensional case, $f \in L_2(\mathbf{R}^d)$, formula (3.17) generalizes as follows:

$$\begin{aligned} f &= \sum_k \alpha_{0k}\ \Phi_{0k} + \sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \beta_{jk}^{(l)}\ \Psi_{jk}^{(l)} \\ \alpha_{jk} &= \langle f, \Phi_{jk} \rangle, \quad \beta_{jk}^{(l)} = \langle f, \Psi_{jk}^{(l)} \rangle\ , \end{aligned} \tag{3.18}$$

where the Φ_{0k}'s and $\Psi_{jk}^{(l)}$'s are the basis functions defined in (3.9).

3.3 Wavelets and functional spaces

We first state a result [55] [44] concerning functions that satisfy Hölder type conditions. This result then motivates introducing Besov functional spaces. Recall, that a function f is called Hölder continuous with exponent s at point x_0, written $f \in \mathcal{C}^s_{x_0}$, if there is a polynomial P of degree at most $\lfloor s \rfloor$ such that [4]

$$|f(x) - P(x - x_0)| \leq C|x - x_0|^s\ .$$

If f is Hölder continuous, with exponent s at x_0, then there exists $C < \infty$ such that, for $j > 0$,

$$\max_{\{k\,:\,x_0 \in \text{ supp } \psi_{jk}\}} \langle f, \psi_{jk} \rangle\ \leq\ C\ 2^{-j(s+d/2)}\ . \tag{3.19}$$

Conversely, if (3.19) holds and f is known to be $\mathcal{C}^{\varepsilon}_{x_0}$ for some $\varepsilon > 0$, then

$$|f(x) - P(x - x_0)| \leq C|x - x_0|^s \log \frac{2}{|x - x_0|}\ .$$

This result states that local smoothness of Hölder type can be characterized with the vanishing rate of the wavelet coefficients in the neighbourhood of this point. This property is specific to the wavelet transform, and does not hold for other orthogonal bases. This remark also motivates introducing Besov spaces of functions.

3.3.1 Besov spaces as spaces of smooth functions with localized singularities

Smooth functions with sparse singularities are typically encountered in nonlinear systems, e.g., in mechanical and chemical systems. As we shall see, Besov spaces are spaces

- of smooth functions with possibly localized singularities,
- in which norms are easily evaluated using wavelet coefficients.

[4] recall that $\lfloor s \rfloor$ denotes the largest integer $\leq s$.

For the sake of clarity we consider only compactly supported functions f: supp $f \subseteq [0,1]^d$, though all the definitions below can be generalized for noncompact and multi-dimensional case (we recomend [76] and [77] as extremely complete presentations of the current state of the theory of functional spaces).

For $f \in L_1$ and $M \in \mathbf{N}$ we define the local oscillation of order M (or M-oscillation for short) at the point $x \in [0,1]$ by

$$\mathrm{osc}_M f(x,t) \quad \triangleq \quad \inf_P \frac{1}{t^d} \int_{|x-y|<t} |f(y) - P(y)| \, dy \ , \tag{3.20}$$

where the infimum is taken over all polynomials P of degree less than or equal to M. This quantity measures the quality of local fit of f by polynomials on balls of radius t.

Select $p, q > 0$, $s > d(p^{-1} - 1)$, and take $M = \lfloor s \rfloor$. The following set of functions:

$$\mathcal{B}^s_{pq} = \left\{ f \in L_{1 \wedge p} \, : \, \|f\|_{\mathcal{B}^s_{pq}} = \|f\|_p + \left(\sum_{j=1}^{\infty} (2^{j\,s} \| \, \mathrm{osc}_M f(x, 2^{-j}) \|_p)^q \right)^{1/q} < \infty \right\} \tag{3.21}$$

(with the usual modification for p or $q = \infty$) is identical to the *Besov spaces* of functions [6], and it is shown in [76] that $\| \cdot \|_{\mathcal{B}^s_{pq}}$ is equivalent to the classical Besov norm.

Comments:

1. The triple parametrization using s, p, and q provides a very accurate characterization of smoothness properties. As usual for Hölder or Sobolev spaces, index s indicates how many derivatives are smooth. Then, for larger p, $\|f\|_{\mathcal{B}^s_{pq}}$ is more sensitive to details. Finally, index q has no useful practical interpretation, but it is a convenient instrument that serves to compare Besov spaces with the more usual Sobolev spaces $\mathcal{W}^s_p$, as indicated next. It is interesting to notice that the indicator functions of intervals belong to the spaces $\mathcal{B}^s_{s^{-1}\infty}$ for all $s > 0$, this illustrates our claim in the title of this subsection.

2. It can be shown that (cf [76]) for $s \geq 0$, $0 \leq p, q \leq \infty$:

   - The family of Besov spaces includes some more classical spaces. for s non integer, Hölder classes $\mathcal{C}^s = \mathcal{B}^s_{\infty\infty}$, and Sobolev spaces $\mathcal{W}^s_2 = \mathcal{B}^s_{22}$;
   - $\mathcal{B}^s_{pq} \subset \mathcal{B}^{s'}_{p'q'}$ if $p' \geq p$, $q' \geq q$, $s' \leq s - \frac{d}{p} + \frac{d}{p'}$ (strict inequality if $p = \infty$);
   - $\mathcal{B}^0_{pq} \subseteq L_p \subseteq \mathcal{B}^0_{pq'}$ where $q = 2 \wedge p$ and $q' = 2 \vee p$;
   - $\mathcal{B}^s_{pp} \subset \mathcal{W}^s_p \subset \mathcal{B}^s_{p2}$ for $p \leq 2$;
   - $\mathcal{B}^s_{p2} \subset \mathcal{W}^s_p \subset \mathcal{B}^s_{pp}$ for $p \geq 2$.

   In particular, if $s > d/p$, then $\mathcal{B}^s_{pq} \subset \mathcal{C}$.

3.3.2 Approximation in Besov spaces, some general results

We consider the d-dimensional case and supp $f \subseteq [0,1]^d$. *Free knots spline approximations* have been analysed in [63] (theorems 7.3 and 7.4) using Besov spaces. Recall that a function f_n is called the spline function on $[0,1]$ of order k with with n knots if $f_n \in \mathcal{C}^{k-2}$ and there exist points (knots) $0 = x_0 < x_1 \leq x_2 \leq \ldots \leq x_{n-1} \leq x_n = 1$ such that f_n is an algebraic polynomial of degree $k-1$ in each interval (x_{i-1}, x_i). Therefore, a spline is a smooth piecewise polynomial function. One can also consider a d-dimensional spline which is the natural generalisation of the 1-dimensional one.

We now state the so-called *Jackson inequality* for spline approximations. Consider $f \in \mathcal{B}^s_{pq}, p, q > 0$. Then there exists a spline function with n free knots f_n such that the following bound holds:

$$\|f_n - f\|_u \leq C(s,p,q)\ n^{-s/d}\ \|f\|_{\mathcal{B}^s_{p\infty}}\ , \tag{3.22}$$

where u satisfies $s - d/p + d/u > 0$. The converse bound is provided by the *Bernstein inequality:* For any $f \in L_u$, $s - d/p + d/u = 0$, $u < \infty$,

$$\|f\|_{\mathcal{B}^s_{pp}} \leq C(s,p,q)\ \left(1\ +\ n^{s/d}\ \inf_{f_n} \|f - f_n\|_u\right)\ ,$$

where the infimum ranges over the set of spline functions f_n of order $k \geq s+2$ with n free knots. A similar result holds for n-order *rational fraction approximations,* see theorem 8.3 in [63].

In contrast, *linear approximations* perform poorly in Besov spaces. Consider some increasing family $(\mathcal{L}_n)$ of n-dimensional linear subspaces of L_u, $u > p$. Let f_n denote the linear projection of $f \in \mathcal{B}^s_{pq}$ on $\mathcal{L}_n$ using the L_u-norm. Then for any such family $(\mathcal{L}_n)$, there exists a least favorable f such that the following lower bound holds:

$$\|f - f_n\|_u \geq C\ n^{-s'/d}\ \|f\|_{\mathcal{B}^{s'}_{uu}}\ , \tag{3.23}$$

where $s' = s - d/p + d/u$. Consider again the example of the indicator function $f(x) = 1_{\{0 \leq x < a\}}$. Recall that $f \in \mathcal{B}^s_{s^{-1}\infty}$ for any $s > 0$. On the one hand, (3.22) shows that f is approximated using rational fractions with an L_u-error of order $O[\exp(-C\sqrt{n})]$, where n is the order of the rational fraction [63]. Thus rational approximations are very efficient for such a function, and the same is true for splines with free knots. On the other hand, by (3.23), linear approximations of the same function have an L_u-error of order $O(n^{-1/u})$, where n is the dimension of the linear subspace, which is extremely poor for large u. This remark would make rational approximations or splines with free knots very attractive for approximation in Besov spaces. Unfortunately, such approximations are very hard to compute, for example, the optimal positionning of the knots of the spline approximation is very hard to find. It is amazing that *wavelet approximations are as good as spline or rational ones, but are much more easily constructed.* We discuss this next.

3.3.3 Wavelets and Besov spaces : mathematically efficient and practically effective

Let φ be a piecewise continuous scale function satisfying the following conditions :

$$\exists a > 0 \quad : \quad \text{supp } \varphi \in \{|x| \leq a\} \tag{3.24}$$

$$\exists r > s \quad : \quad \varphi \in \mathcal{B}^r_{u\infty} \tag{3.25}$$

We have the following result (c.f. theorem 4 in [72]):

Theorem 6 (Besov norms and wavelet decompositions) *Let $s > d(1/u - 1)$ and φ be a scale function satisfying conditions (3.24) and (3.25). For any $f \in \mathcal{B}^s_{pq}$ define*

$$\|f\|_{spq} = \left(\sum_k |\alpha_k|^p \right)^{1/p} + \left(\sum_{j=0}^{\infty} \left[2^{j(s+d/2-d/p)} \|\beta_{j\cdot}\|_p \right]^q \right)^{1/q} \tag{3.26}$$

and $\|\beta_{j\cdot}\|_p = (\sum_{i,k} |\beta^{(l)}_{jk}|^p)^{1/p}$, see (3.10) (3.18) for the definition of coefficients $\alpha_k = \alpha_{0k}$ and $\beta^{(l)}_{jk}$. Then (3.26) is a equivalent to the norm of Besov space $\mathcal{B}^s_{pq}$, i.e., there exist constants C_1 and C_2, independent of f, such that

$$C_1 \, \|f\|_{\mathcal{B}^s_{pq}} \; \leq \; \|f\|_{spq} \; \leq \; C_2 \, \|f\|_{\mathcal{B}^s_{pq}} \; . \tag{3.27}$$

Theorem 6 states that norms in Besov spaces are suitably evaluated using orthonormal wavelet decompositions. This fact can be used to obtain very efficient approximations.

We now indicate how such a wavelet approximation of f can be constructed. Consider the full wavelet decomposition of f :

$$f(x) \quad = \quad \sum_{k \in \mathbf{Z}} \alpha_{0k} \; \Phi_{0k}(x) + \sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \beta^{(l)}_{jk} \; \Psi^{(l)}_{jk}(x) \; . \tag{3.28}$$

1. Keep the projection of f on the subspace V_0, this corresponds to the left most sum in (3.28). When f and Φ are both compactly supported this requires computing only a fixed amount of coefficients, say m. And then

2. Select in the second (triple) sum those coefficients β_λ, $\lambda = (i, j, k)$ with largest absolute value, denote by Λ the set of the $n - m$ so selected wavelet coefficients. Finally

3. Add $n - m$ detail terms $\beta_\lambda \Psi_\lambda$ to the sum taken in step 1.

This procedure yields the approximation

$$w_n(x) \quad = \quad \underbrace{\sum_k \alpha_{0k} \; \Phi_{0k}(x)}_{\substack{m \text{ coeffs.} \neq 0 \\ (f, \Phi \text{ compact. supp.})}} \quad + \underbrace{\sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \beta^{(l)}_{jk} \; \Psi^{(l)}_{jk}(x)}_{\text{keep the largest } n-m \text{ coeffs.}} \tag{3.29}$$

and the following theorem provides coresponding approximation bounds.

Theorem 7 (DeVore, Jawerth, Popov, [19]) *Consider $f \in \mathcal{B}^s_{pp}$, $s, p > 0$ and $s - d/p + d/u \geq 0$. Let w_n denote the approximation (3.29) of f. If the scale function satisfies conditions (3.24) and (3.25), then*

$$\|f - w_n\|_u \quad \leq \quad C(s,p)\ n^{-s/d}\ \|f\|_{\mathcal{B}^s_{pp}}$$

holds. If, in addition, u satisfies $s - d/p + d/u = 0$, $u < \infty$, and it is a priori known that $f \in L_u$, then the following converse bound holds.

$$\|f\|_{\mathcal{B}^s_{pp}} \quad \leq \quad C(s,p,q)\ \left(1 \ + \ n^{s/d}\ \|f - w_n\|_u\right) \ .$$

This result is very interesting for us. It implies that, in the wavelet decomposition of a function $f \in \mathcal{B}^s_{pq}, p < 2$, only a small number of coefficients are important, and the other ones can be neglected. Consider once more our example $f(x) = 1_{\{0 \leq x < a\}}$. Consider the wavelet decomposition of this function using a compactly supported wavelet $\psi(x)$ such that $\int \psi(x)dx = 0$. It is evident that the coefficient β_{jk} vanishes for any wavelet $\psi_{jk}(x)$ which does not across the (local) singularities of f. Thus if we consider the projection of f on the subspace V_j, only $O(j)$ coefficients of the decomposition significantly differ from zero (among 2^j potential candidates).

Discussion. At this point we have the requested background for understanding how to perform wavelet based estimation. Roughly speaking, the crux is the following. The function $f \in \mathcal{B}^s_{pq}$ to be estimated can be *approximated* using expansion w_n in (3.29) with n terms. This is achieved with a rate of $O(n^{-s/d})$. Then coefficients α_k and β_λ in (3.29) are estimated via empirical means based on N noisy observations, exactly as for the projection estimates in Section 2, formula (2.14). The mean square error on the estimate of each coefficient is $O(1/N)$. Thus the total mean square error of the estimate will be, as usual, the sum of the stochastic part and of the bias due to the approximation error : this yields $O(n/N) + O(n^{-2s/d})$. The optimal choice for n balances these two terms : $n = N^{\frac{1}{2s+d}}$. This choice for n yields a quadratic error of order $N^{-\frac{2s}{2s+d}}$ (independent of p, q). As we shall see, this is the typical minimax rate of convergence on Besov spaces. Thus we might be ready to deduce that wavelet estimators are minimax optimal in Besov spaces. Unfortunately, the set Λ of "important" coefficients in truncation (3.29) is *not* known a priori when noisy data sets are at hand for estimation. Thus some kind of hypothesis testing problem must be solved in order to obtain the optimal approximation. This adds to the estimation problem a nice stochastic flavour. We address this point in the next chapter.

Chapter 4

Wavelets : their use in nonparametric estimation

We consider here some simple results concerning the estimation of a regression function or a density $f : \mathbf{R}^d \to \mathbf{R}$, and we assume f to be compactly supported (supp $f \subseteq [0,1]^d$). For the sake of simplicity we measure the estimation error in L_2-norm. Similar results were proved for a general d-dimensional case and a variety of error measures, which includes, for instance, L_p-norms for $0 < p \leq \infty$ (see the references at the end of the section). We successively discuss the problems of non-parametric regression and density estimation.

4.1 Wavelet shrinkage algorithms

Non-parametric regression. Assume a N-sample of input/output observations of the following system are available:

$$Y_i = f(X_i) + w_i \ ,$$

where (X_i) and (w_i) are i.i.d. sequences of random variables, X_i is *uniformly distributed* on $[0,1]^d$ and $Ew_i = 0$, $Ew_i^2 \leq \sigma_w^2$. These assumptions are introduced for the sake of simplicity. They can be weakened, in particular the (unusual) assumption that X is uniformly distributed can easily be relaxed, see [15], this would introduce additional burden to our presentation, however.

For $f \in L_2$, recall the wavelet expansion

$$f(x) \quad = \quad \sum_{k \in \mathbf{Z}} \alpha_{0k} \ \Phi_{0k}(x) + \sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \beta_{jk}^{(l)} \ \Psi_{jk}^{(l)}(x) \ , \tag{4.1}$$

where

$$\alpha_{0k} = \int f(x)\Phi_{0k}(x)dx \quad \text{and} \quad \beta_{jk}^{(l)} = \int f(x)\Psi_{jk}^{(l)}(x)dx \ . \tag{4.2}$$

To construct an estimate of f a first idea consists in using the law of large numbers and replacing, in expansion (4.1), the coefficients α_k and $\beta_{jk}^{(l)}$ by their empirical estimates

$$\widehat{\alpha}_{0k}(N) = \frac{1}{N}\sum_{i=1}^{N} Y_i\Phi_{0k}(X_i) \quad \text{and} \quad \widehat{\beta}_{jk}^{(l)}(N) = \frac{1}{N}\sum_{i=1}^{N} Y_i\Psi_{jk}^{(l)}(X_i) \ . \tag{4.3}$$

Note that the assumption that input X is uniformly distributed has been used at this point.

Density estimation. Assume independent observations $X_1, \ldots, X_N$ of some random variable X with unknown density $f(x)$ are available. Again f can be expanded using (4.1) (4.2). But it turns out that

$$\alpha_{0k} = \int f(x)\Phi_{0k}(x)dx \quad = \quad \mathbf{E}_f\Phi_{0k}(X_i)$$

where $\mathbf{E}_f$ denotes expectation with respect to density f, and the same holds for the β's. Thus empirical estimates of the wavelet coefficients α_k and β_{jk} are given by

$$\widehat{\alpha}_{0k} = \frac{1}{N}\sum_{i=1}^{N} \Phi_{0k}(X_i) \quad \text{and} \quad \widehat{\beta}_{jk}^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \Psi_{jk}^{(l)}(X_i) \ . \tag{4.4}$$

Thus both non-parametric regression and density estimation are faced with the same issue : in formulas (4.3) and (4.4), they may not even be available X_i's within the support of many of the Φ's and Ψ's ! We shall now discuss this key point for the case of density estimation.

Obviously, in order to compute the empirical coefficient $\widehat{\beta}_{jk}^{(l)}$, we need that at least several observations X_i hit the support of $\Psi_{jk}^{(l)}(x)$. Statistical laws of *loglog* type guarantee that this would generically hold for scales that are not too fine. More specifically, for $j \leq j_{\max}$, where

$$\frac{N}{\ln N} \leq 2^{d\,j_{\max}} \leq \frac{2N}{\ln N}$$

Thus we bruteforce set $\widehat{\beta}_{jk}^{(l)} = 0$ for $j > j_{\max}$. At this point we have built an estimator of the linear projection type, as in the case of Fourier series in section 2.1. Since these estimators are linear, we cannot expect them to be efficient for Besov spaces [46].

A first proposal. Our first attempt to construct an "interesting estimate" is, following the intuition at the end of the previous section, to keep a properly chosen number of coefficients with largest absolute values, and set the others to zero. More precisely, let us consider the

set $\widehat{\Lambda}_n$ of pairs $\lambda = (j,k)$ corresponding to the n estimated wavelet coefficients $\widehat{\beta}_{jk}^{(l)}$ with largest absolute values. We construct the estimate $\widehat{f}_N$ as follows:

$$\widehat{f}_N(x) \quad = \quad \underbrace{\sum_k \widehat{\alpha}_{0k}\ \Phi_{0k}(x)}_{\substack{m \text{ coeffs.} \neq 0 \\ (f, \Phi \text{ compact. supp.})}} \quad + \underbrace{\sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \widehat{\beta}_{jk}^{(l)}\ \Psi_{jk}^{(l)}(x)}_{\text{keep the largest } n-m \text{ coeffs.}} \ . \tag{4.5}$$

The following result can be proved about estimate (4.5) (see (3.21) for the definition of the Besov spaces) :

Theorem 8 *Let $f \in \mathcal{B}^s_{p\infty}$ with $s \geq d/p$, $\|f\|_\infty < \infty$. If $n = N^{1/(2s+d)}$ is selected in (4.5), then*

$$E\|\widehat{f}_N - f\|_2^2 = O\left(\frac{\ln N}{N}\right)^{\frac{2s}{2s+d}} \ . \tag{4.6}$$

The idea of the proof of theorem 8 is quite intuitive and typical for wavelet estimators. We follow the argument at the end of the previous chapter with the only following difference : since no information is available about the distribution of the error $|\widehat{\beta}_\lambda - \beta_\lambda|$ for $\lambda \in \widehat{\Lambda}_n$, we take a cautious upper bound for it :

$$\mathbf{E}\ |\widehat{\beta}_\lambda - \beta_\lambda|^2\ 1_{\{\widehat{\beta} \neq 0\}} \quad \leq \quad \mathbf{E}\ \sup_{i,j,k} |\widehat{\beta}_{jk}^{(l)} - \beta_{jk}^{(l)}|^2 = O\left(\frac{\ln N}{N}\right),$$

which explains the extra logarithmic factor in (4.6).

The final solution. Note that n in Theorem 8 depends on s, which is generally unknown. Hence, to complete the estimation algorithm, we need a method to estimate our model order n. Though Generalised Cross-Validation type techniques could be used, we prefer a somewhat different estimation approach developped by D. Donoho, I. Johnstone, G. Kerkyacharian and D. Picard (see the references below). It uses simple thresholding rules [1] :

$$\widetilde{\beta}_{jk}^{(l)} \quad = \quad \widehat{\beta}_{jk}^{(l)}\ 1_{\{|\widehat{\beta}| \geq \lambda_j\}} \tag{4.7}$$

where λ_j is a threshold parameter, so we set

$$\widehat{f}(x) \quad = \quad \sum_k \widehat{\alpha}_{0k}\ \Phi_{0k}(x) + \sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}^d} \sum_{l=1}^{2^d-1} \widehat{\beta}_{jk}^{(l)}\ 1_{\{|\widehat{\beta}_{jk}^{(l)}| \geq \lambda_j\}} \Psi_{jk}^{(l)}(x) \ . \tag{4.8}$$

[1] we consider here the so called "hard thresholding", meanwhile, other rules can also be studied, for example the "soft thresholding" [26]. See also the discussion in [17].

In other words, in expansion (4.1), we keep those empirical estimates of wavelet coefficients which exceed some properly selected threshold. How this threshold should be selected is provided by the following result :

Theorem 9 ([28] and [29]) *Let $f \in \mathcal{B}^s_{p\infty}$ with $s \geq d/p$, $\|f\|_\infty < \infty$. Select $\lambda_j = \lambda = \sqrt{\frac{C \ln N}{N}}$, with an appropriate $C < \infty$. Then*

$$E\|\widehat{f}_N - f\|_2^2 = O\left(\frac{\ln N}{N}\right)^{\frac{2s}{2s+d}} .$$

The constant C in the expression for the threshold parameter λ is a sort of an "hyperparameter" of the procedure, it can be easily estimated, see [17] and [28] for related discussions. Note that the estimator $\widehat{f}_N$ is adaptive because *it does not require prior knowledge of the regularity parameter.*

Discussion.

- Theorem 9 has the following intuitive explanation. As already mentioned, Besov classes $\mathcal{B}^s_{pq}$ for $p < 2$ have a special structure : a relatively small number of "important" wavelet coefficients are sufficient for obtaining a good function approximation. In the wavelet decomposition $(\widehat{\alpha}_k, \widehat{\beta}^{(l)}_{jk})$ using noisy data, all coefficients are "contaminated" by noise. A Central Limit theorem argument suggests that this noise is approximately Gaussian with zero mean and variance $O(1/N)$. Thus loglog law implies that the maximal error in the estimates has magnitude given by

  $$\max_{j,k} |\widehat{\beta}^{(l)}_{jk} - \beta^{(l)}_{jk}| \approx \sqrt{\frac{2 \ln N}{N}} .$$

  Thus when small (according to threshold λ in Theorem 9) coefficients are shrinked to zero, noise is cancelled with very high probability. On the other hand, coefficients exceeding this threshold are likely to be significantly different from zero. This property of thresholding explains another useful feature of the estimator: the estimate $\widehat{f}_N$ has the same regularity as the unknown function f to be estimated (cf. discussion in [29]).

- Let us now consider again our example of estimating the regression function or density $f(x) = 1_{\{0 \leq x < a\}}$. Theorem 9 states that the mean square rate of convergence of the wavelet estimator for any bounded function $f \in \mathcal{B}^s_{s^{-1}\infty}$ is very close to $O(N^{-1})$, which is nearly as good as the "parametric" rate of convergence, though the function we estimate is not even continuous. Let us compare the results above with the lower rate of convergence for this problem obtained in [60]. Using the comments 2. of section 3.3.1, the following lower bound is a direct corollary of the results of [60] which were originally formulated in terms of Sobolev spaces :

  $$\inf_{\widehat{f}_N} \sup_{f \in \mathcal{B}^s_{pq}} E\|\widehat{f}_N - f\|_2 \geq C N^{-2s/(2s+d)} \tag{4.9}$$

for any estimator $\widehat{f}_N$. As compared to (4.9), there is an extra logarithmic factor in the upper bound of theorem 9. In the more subtle construction presented in [25], this logarithmic factor is eliminated (and even a precise minimax constant is obtained) in the case of Gaussian noises and deterministic design (observations are $x_i = i/N$, $i = 1, \ldots, N$). In [27] a cross-validation procedure is proposed to adapt the optimal algorithm to unknown smoothness. Finally, in [17] the authors of this paper showed that properly selecting the threshold λ for shrinking provides the optimal rate of convergence (without a logarithmic factor). An adaptive version of this algorithm is developped in [45].

4.2 Practical implementation of wavelet estimators

We now move to the practical implementation of wavelet estimators. We propose two versions of it which differ in the way the empirical estimates of the wavelet coefficients $\widehat{\alpha}_{jk}$ and $\widehat{\beta}_{jk}$ are computed. The first one, we call it *direct realization*, is based on the explicit formulae (4.3) and (4.4) for empirical coefficients. The second one, called *fast realization* procedure, relies on the Quadrature Mirror Filters (QMF) presented in section 3.2.2.

Direct wavelet estimation procedure for an N-sample length (put $Y_i \equiv 1$ for density estimation): the WSA procedure. (Recall that the assumption that X is uniformly distributed is required for the case of regression.)

1. Select $j_{\max}$ scales for the wavelet expansion, where

$$\frac{N}{\ln N} \leq 2^{d j_{\max}} \leq \frac{2N}{\ln N}$$

2. For $j \leq j_{\max}$ compute the empirical estimates

$$\widehat{\alpha}_k = \frac{1}{N} \sum_{i=1}^{N} Y_i \; \Phi_{0k}(X_i) \; , \quad \widehat{\beta}_{jk}^{(l)} = \frac{1}{N} \sum_{i=1}^{N} Y_i \; \Psi_{jk}^{(l)}(X_i) \; . \tag{4.10}$$

3. Shrink these estimates according to

$$\widetilde{\beta}_{jk}^{(l)} \quad = \quad \widehat{\beta}_{jk}^{(l)} \; 1_{\{|\widehat{\beta}_{jk}^{(l)}| \geq \lambda_j\}} \; , \tag{4.11}$$

   where λ_j is a properly selected threshold (cf. theorem 9).

4. The final estimate is given by

$$\widehat{f}_N(x) \quad = \quad \sum_k \widehat{\alpha}_k \; \Phi_{0k}(x) + \sum_{i,j,k} \widetilde{\beta}_{jk}^{(l)} \; \Psi_{jk}^{(l)}(x) \; . \tag{4.12}$$

This procedure for nonparametric regression can be extended to the case in which X is *not* uniformly distributed over $[0,1]^d$, and has density $g(x)$. In this case, we have

$$\widehat{\alpha}_k = \frac{1}{N}\sum_{i=1}^{N} Y_i\ \Phi_{0k}(X_i)\ \approx\ \int f(x)\ \Phi_{0k}(x)\ g(x)\ dx\ =\ \int [fg](x)\ \Phi_{0k}(x)\ dx$$

and similarly for the $\widehat{\beta}_{jk}^{(l)}$'s. Thus applying WSA to estimate regression function f (with the Y_i in the empirical estimates) as if X was uniformly distributed yields in fact an estimate $\widehat{[fg]}_N$ of $[fg]$. From this remark the following procedure follows.

1. apply WSA to estimate density g (without the Y_i in the empirical estimates): this yields $\widehat{g}$;
2. apply WSA to estimate regression function f (with the Y_i in the empirical estimates) as if X was uniformly distributed: this yields $\widehat{f}_{\text{uniform}}$;
3. the final estimate is $\widehat{f} = \widehat{f}_{\text{uniform}}/\widehat{g}$.

COMMENT : The above *direct* estimate has some drawbacks (we consider only the computational aspect for a moment). First, we know that there is no closed form for the scale function Φ or wavelet Ψ, thus in order to compute $\widehat{\alpha}_{jk}$ and $\widehat{\beta}_{jk}$ we would have to compute and store the values of Φ and Ψ on a fine grid, which is prohibitive. Second, we would like to take advantage of the fast QMF algorithms of section 3.2.2 for computing orthonormal wavelet decompositions. We can not apply these algorithms directly on the data, since the available observations $X_1, \ldots, X_N$ are randomly sampled and do not form a regular grid. To circumvent this difficulty, we preprocess the observations to obtain the empirical coefficients $\widehat{\alpha}_{j_{\max},k}$ at the finest resolution level $j_{\max}$; then we can apply the QMF algorithms of section 3.2.2 to compute the coefficients at coarser scales. The proposed procedure is close to the *empirical wavelet transform* or *hybrid transform*, studied in section 5 of [24], mathematical details can be found in [16]. We assume that the function f is supported on $[0,1]^d$.

Fast wavelet estimator (X does not need to be uniformly distributed).

1. **(preprocessing)** Select again $j_{\max}$ such that

$$\frac{N}{\ln N} \le 2^{dj_{\max}} < \frac{N}{\ln N}\ .$$

Let $k = (k_1, \ldots, k_d)^T$ be a multi-index, consider the bin

$$\Delta_k\ =\ \left[2^{-j_{\max}}k_1, 2^{-j_{\max}}(k_1+1)\right]\ \times \ldots \times\ \left[2^{-j_{\max}}k_d, 2^{-j_{\max}}(k_d+1)\right]\ .$$

For density estimation we first take the empirical probability of bin Δ_k (recall that Δ_k has volume $2^{-dj_{\max}}$), this yields :

$$\widetilde{f}_{N,k}\ =\ 2^{dj_{\max}}\ \frac{1}{N}\sum_{i=1}^{N} 1_{\{X_i \in \Delta_k\}}\ , \quad \text{and then}\ \ \widehat{\alpha}_{j_{\max},k}\ =\ 2^{dj_{\max}/2}\ \widetilde{f}_{N,k}\ .$$

For nonparametric regression, similarly, compute

$$\widetilde{f}_{N,k} \;=\; \frac{\sum_{i=1}^{N} Y_i \; 1_{\{X_i \in \Delta_k\}}}{\sum_{i=1}^{N} 1_{\{X_i \in \Delta_k\}}} \;, \quad \text{and then} \;\; \widehat{\alpha}_{j_{\max},k} \;=\; 2^{dj_{\max}/2} \; \widetilde{f}_{N,k} \;.$$

At this point we have constructed synthetic input-output pairs, where the input is the considered bin and output is the associated $\widehat{\alpha}_{j_{\max},k}$ estimate. Getting the full wavelet expansion is then performed by applying to these synthetic data the QMF fast formulae (3.11), (3.12).

2. **(QMF filtering)** Use the multi-dimensional version of filters (3.11), (3.12) to compute $\widehat{\alpha}_{jk}$, $\widehat{\beta}^{(l)}_{jk}$, $j = 0, \ldots, j_{\max} - 1$, $l = 1, \ldots, 2^d - 1$:

$$\begin{aligned} \widehat{\alpha}_{jk} &= \sum_i \overline{h}_{i-2k} \; \widehat{\alpha}_{j+1,i} \\ \widehat{\beta}^{(l)}_{jk} &= \sum_i \overline{g}^{l}_{i-2k} \; \widehat{\alpha}_{j+1,i} \;. \end{aligned}$$

3. Shrink the estimates $\widehat{\beta}^{(l)}_{jk}$ according to

$$\widetilde{\beta}^{(l)}_{jk} \;=\; \widehat{\beta}^{(l)}_{jk} \; 1_{\{|\widehat{\beta}^{(l)}_{jk}| \geq \lambda_j\}} \;,$$

where λ_j is a properly selected threshold (cf. theorem 9).

4. Use the "inverse" filter (3.15) to obtain $\widetilde{\alpha}_{j_{\max},k}$:

$$\widetilde{\alpha}_{jk} \;=\; \sum_{il} h_{k-2i} \; \widetilde{\alpha}_{j-1,i} + g^{l}_{k-2i} \; \widetilde{\beta}^{(l)}_{j-1,i} \;. \tag{4.13}$$

5. Finally set

$$\widehat{f}_N(2^{-j_{\max}}k) \;\stackrel{\Delta}{=}\; \widetilde{f}_{N,k} \;=\; 2^{-dj_{\max}/2} \; \widetilde{\alpha}_{jk} \;.$$

In this way we obtain estimates of $f(2^{-j_{\max}}k)$. If this accuracy is not sufficient, it is possible to interpolate $\widetilde{f}_N$ at a finer grid by applying upsampling (4.13), using the filters that are biorthogonal to those associated with the Haar basis (see, chapter 8 in [14], or [24]).

Chapter 5

A wavelet network for practical system identification

The estimation procedure described in the previous section may not be effective for X of higher dimension, and sparse input data sets for training. In this chapter we attempt to cope with highly-dimensional problems and bad data sampling using an alternative technique of wavelet estimation. We present here a method for constructing estimators with non orthogonal wavelets, the corresponding software is available [89]. We investigate problem 1 of Section 1.2 in the case of additive noise, i.e., we suppose that the pair of random variables X, Y satisfies

$$Y = f(X) + e \ , \tag{5.1}$$

where $f(x) : \mathbf{R}^d \mapsto \mathbf{R}$ and e is some noise of zero mean and independent of X. We want to estimate f based on a sample of size N that we shall refer to as the *training data set*: $\mathcal{O}_1^N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$. We are particularly interested in training with sparse data sets. Sparse data often occur in classification problems and in the modeling of control systems, where available data can be relatively few as compared to the dimension of input X. Throughout this chapter, φ shall denote a radial wavelet as defined in Theorem 3, thus we are *not* using orthonormal wavelets.

5.1 Adaptive dilation/translation sampling

We present here a result which can be regarded as a theoretical justification of the techniques in this chapter. Note that in the orthonormal wavelet expansion

$$f(x) = \sum_k \alpha_{0k} \Phi_{0k}(x) + \sum_{ljk} \beta_{jk}^{(l)} \Psi_{jk}^{(l)}(x) \ ,$$

the dilation and translation parameters – 2^{-dj} and k do not depend on the function to expand and only the linear weights α_{jk} and $\beta_{jk}^{(l)}$ depend on f. Suppose that we construct a wavelet "basis" with dilations and translations depending on the function f. The wavelet expansion of f using these basis functions is expected to use less wavelets, and thus we expect it to be more convenient for estimation purposes. To obtain such a basis *we discretize the continuous wavelet transform* (3.3) (see section 3.1).

We first recall the following algorithm proposed in [18]. Consider the continuous wavelet transform (3.3), which we rewrite as

$$\begin{aligned} f(x) &= \int u(a,t)\ \varphi(a(x-t))\ a^{d-1/2}\ da\ dt \\ &= \int \varphi(a(x-t))\ \mathrm{sign}(u(a,t))\ a^{(d-1)/2}\ |u(a,t)|\ da\ dt \\ &= \frac{1}{C} \int \varphi(a(x-t))\ \mathrm{sign}(u(a,t))\ w(a,t)\ da\ dt \end{aligned}$$

where we have renormalized $u(a,t)$ by a constant factor C so that the function $w(a,t) = Ca^{(d-1)/2}\ |u(a,t)|$ can be considered as a probability density. Then we draw n independent random samples $(a_i,t_i)_{i=1,\ldots,n}$ from distribution with density $w(a,t)$. Then we build

$$f_n(x) = \frac{1}{n} \sum_{i=1}^{n} a_i^{d/2}\ \varphi(a_i(x-t_i))\ \mathrm{sign}(u(a_i,t_i))\ , \tag{5.2}$$

which, thanks to the law of large numbers, converges to the true wavelet transform. Some faster implementations of this algorithm are given in [18]. Improving this estimate by some "bootstrapping" like technique, yields the following approximation result.

Theorem 10 ([18]) *φ is any radial wavelet function such that there exists a related radial function ψ which satisfies condition (3.1). Let p, μ, l, ρ be real numbers satisfying*

$$1 < p < \left(1 - \frac{\rho - l}{d}\right)^{-1}, \qquad \mu = \min\left(1 - \frac{1}{p}\ ,\ \frac{1}{2}\right)$$

and f be a function of the Sobolev space $\mathcal{W}_1^\rho(\mathbf{R}^d)$; then, for any $n > 0$ there exists a function f_n of the form

$$f_n(x) = \sum_{i=1}^{n} u_i\ \varphi(a_i(x-t_i)) \tag{5.3}$$

such that

$$\|f_n - f\|_{\mathcal{W}_p^l} \leq C n^{-\mu} \|f\|_{\mathcal{W}_1^\rho}\ .$$

In particular, if $\rho > d/2$ then

$$\|f_n - f\|_2 \leq n^{-1/2} C \|f\|_1^\rho\ .$$

COMMENT : Theorem 10 provides us with an upper bound for the rate of approximation when adaptive dilation/translation sampling is used to discretize the continuous wavelet transform. We should compare this rate with rates of convergence for approximations based on *fixed* dilation/translation sampling. For example, the following theorem is proved in [18] :

Theorem 11 *Let $p = 2$ and $\rho = d/2 + \varepsilon$, $\varepsilon > 0$. For a collection $h_1, \ldots, h_n$ of basis functions*[1]*, consider the error*

$$V_n = \inf_{h_1,\ldots,h_n} \sup_{\|f\|} \|f - \mathrm{span}\{h_1, \ldots, h_n\}\|_2 \ ,$$

where span$\{\ldots\}$ *denotes the linear space spanned by the listed functions, and the supremum is taken over the unit ball $\mathcal{B} = \{f : \|f\|_1^\rho \leq 1\}$ of the Sobolev space $\mathcal{W}_1^\rho$. Then there exists a universal constant C such that, for any fixed basis $h_1, \ldots, h_n$,*

$$V_n \geq C n^{-\varepsilon/d} \ .$$

The result of the theorem implies that for any *fixed* basis $h_1, \ldots, h_n$ and any set of $\alpha_1, \ldots, \alpha_n$, there are "worst functions" f for which a projection approximation $f_n^h(x)$ of the form

$$f_n^h(x) = \sum_{i=1}^{n} \alpha_i h_i$$

converges much slower than the approximation (5.3). Note that this is not in contradiction with the optimality of wavelet shrinkage procedures, since shrinking coefficients in the wavelet expansion makes the estimator to be nonlinear.

5.2 The wavelet network and its structure

Though the above adaptive dilation/translation sampling algorithm provides us with a good basis, its implementation using Monte-Carlo technique is of prohibitive computational cost. We rather implement adaptive sampling in a different way, by combining regressor selection and backpropagation algorithms in order to find good dilations and translations. The resulting estimator is called *wavelet network*. Related works have been reported in [91, 87, 88]. We refer the reader to [91] for heuristic comparisons between neural and wavelet networks. For any wavelet function $\varphi : \mathbf{R}^d \to \mathbf{R}$, the wavelet network is written as follows

$$f_n(x) = \sum_{i=1}^{n} u_i \varphi(a_i \star (x - t_i)) \tag{5.4}$$

[1] one can take, for instance, the trigonometric basis on $[0,1]^d$, or a truncated wavelet basis with fixed dilation and translation sampling.

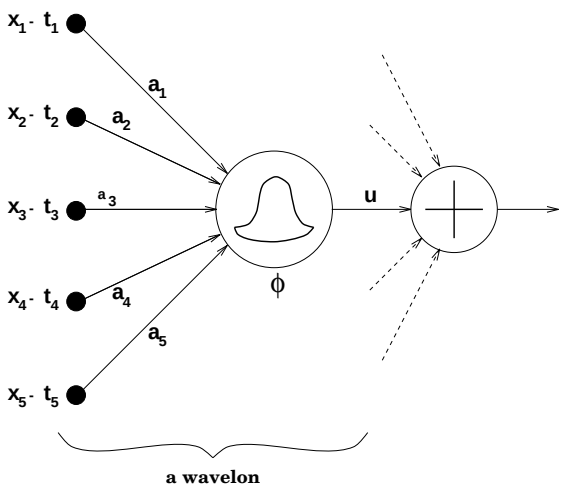

Figure 5.1: *The wavelet network. A waveleon is shown, which corresponds to one term* $\varphi(a_i \star (x - t_i))$. *Dashed arrows figure output connections to other wavelons.*

where $u_i \in \mathbf{R}, a_i \in \mathbf{R}^d, t_i \in \mathbf{R}^d$, and "$\star$" means component-wise product of two vectors. Note that we could have used scalar dilation parameters a_i, but we prefer vectorial dilation parameters because they considerably increase the flexibility of network (5.4) at a reasonable price. The strcuture of the wavelet network is depicted in Figure 5.1. In this chapter we present an efficient comprehensive method for wavelet network training. The following is an outline of this method.

1. Construct a library W of dilated/translated versions of a given wavelet φ. This library W is adapted to the available training data set, by selecting a subset from all dilated/translated versions of φ on a regular grid. This technique makes it feasible to build the library W even for significantly large input dimension when the training data are sparse.

2. Not all wavelets from library W are useful in fitting f from noisy data, however. This leads to the problem of selecting the best wavelet regressors among W. Three heuristic methods will be proposed for this. When the regressors are conveniently selected, fitting model (5.4) amounts to identifying the u_i coefficients, which is a standard least squares estimation problem.

3. Steps 1 and 2 above yield a fast training procedure. The result can still be further improved by subsequently applying an iterative backpropagation algorithm with steps 1 and 2 as fast initialization. In fact, since initialization was good, a faster Newton procedure can be used.

More details of each step are given below.

5.3 Constructing the wavelet library W

First we should build a library W of wavelets which will be considered as candidates of regressors. We have to restrict ourselves to a finite set of regressor candidates, in order to apply regressor selection algorithms. Naturally W is chosen to be a subset of the continuously parameterized family $\{\varphi(a(x-t)) : a \in \mathbf{R}^+, t \in \mathbf{R}^d\}$. The choice of W is in principle the same problem as discretizing the continuous wavelet reconstruction (3.3) to obtain the discrete reconstruction (3.4). The standard discretization is a regular lattice:

$$\{\varphi(a_0^n x - m t_0) : n \in \mathbf{Z}, m \in \mathbf{Z}^d\} \tag{5.5}$$

where $a_0, t_0 > 0$ are two scalar constants defining the discretization step sizes for dilation and translation, respectively. Typically we take a *dyadic lattice*. Now the countable family (5.5) should be truncated into a finite set. Usually we only want to estimate $f(x)$ on a compact domain $D \subset \mathbf{R}^d$ and the wavelet function $\varphi(x)$ is chosen to have compact or rapidly vanishing support, therefore we can replace in (5.5) $m \in \mathbf{Z}^d$ by $m \in S_t$ with a finite set $S_t \subset \mathbf{Z}^d$; on the other hand, $n \in \mathbf{Z}$ should be replaced by $n \in S_a$ with a finite set $S_a \subset \mathbf{Z}$ corresponding to the "desired" resolution levels of the estimation. In practice, 4 or 5 consecutive dilation levels are usually sufficient, with the largest wavelet scale corresponding to the size of D, the compact domain on which f is to be estimated. After such a truncation being performed, family (5.5) is replaced by

$$\{\varphi(a_0^n x - m t_0) : n \in S_a, m \in S_t(n)\} \tag{5.6}$$

Note that the cardinality of this wavelet library grows exponentially with the dimension d. The following procedure is used to overcome this curse of dimensionality when the training data are sparse : scan the training data set $\mathcal{O}_1^N$, for each sample point in $\mathcal{O}_1^N$, determine the wavelets in (5.6) whose supports[2] contain this data point; add these wavelets to W if they have not figured in it. With this method, the dimension d is not a critical factor of complexity, since family(5.6) does not need to be actually created. For sparse training data set, this method allows to handle problems of relatively large input dimension d. In particular, if the supports of the wavelets are approximated by hyper-cubes in $\mathbf{R}^d$, this method is easily implemented. From now on we denote by W the resulting *library of wavelet regressor candidates.* For computational convenience we normalize the wavelets and get the library W composed of the wavelets :

$$\begin{aligned} \varphi_i(x) &= \alpha_i \ \varphi(a_i(x - t_i)), \quad i = 1, \ldots, L \\ \alpha_i &= \left(\sum_{k=1}^{N} [\varphi(a_i(x_k - t_i))]^2 \right)^{-\frac{1}{2}}, \end{aligned}$$

[2] For non compactly supported but rapidly vanishing wavelets, the term "support" should be interpreted in an approximative way as some domain around the center of the wavelet.

where L is the number of elements in W, a_i, t_i correspond to the dilation and translation parameters a_0^n and $a_0^{-n} m t_0$ of the wavelet φ_i, and α_i is the normalizing factor. The numbering order with i is arbitrary.

5.4 Selecting best wavelet regressors

The problem of regressor selection is to select a number $M \leq L$ of wavelets, the "best" ones from W for building the regression.

$$f_M(x) = \sum_{i \in I} u_i \varphi_i(x) \tag{5.7}$$

where I is a M-elements subset of the index set $\{1, 2, \ldots, L\}$. This is a classical problem in regression analysis [30]. Let $\mathcal{I}_M$ be the set of all the M-elements subsets of $\{1, 2, \ldots, L\}$. For any $I \in \mathcal{I}_M$, the optimal linear weights u_i of (5.7) are found using the least squares method. Then the question is how to choose $I \in \mathcal{I}_M$ which minimizes the averaged square residuals

$$J(I) \quad = \quad \min_{\{u_i : i \in I\}} \frac{1}{N} \sum_{k=1}^{N} \left(Y_k - \sum_{i \in I} u_i \varphi_i(X_k) \right)^2 . \tag{5.8}$$

Determining the optimal number M should be performed using Generalized Cross Validation, cf. Section 2.1.2. For given M, selecting the M optimal regressors from W must be performed via exhaustive search which may involve massive computations. To overcome this difficulty, three different heuristics are proposed instead, details can be found in the appendix.

Residual based selection (RBS). The idea of this method is to select, for the first stage, the wavelet in W that best fits the observations $\mathcal{O}_1^N$, then repeatedly select the wavelet that best fits the residual of the fitting of the previous stage. In the literature of the classical regression analysis, it is considered as an simple, but not quite effective method, for example in [30] where it is called *stagewise regression procedure*. For classical regressions the number of regressor candidates is usually small, hence alternative more complicated and more effective procedures are preferred. In our situation the number of regressor candidates may reach several hundreds or even more, the computational efficiency becomes more important and the simple residual based selection should be a first choice. Recently it has also been used in the matching pursuit algorithm of S. Mallat and Z. Zhang [53] and the adaptive signal representation of S. Qian and D. Chen [68]. This procedure is described in Appendix A.1

Stepwise selection by orthogonalization (SSO). The idea of this method is to select, for the first stage, the wavelet in W which best fits the observations $\mathcal{O}_1^N$, then repeatedly select the wavelet that best fits $\mathcal{O}_1^N$ while working together with the previously selected wavelets. This method has been used in radial basis function (RBF) networks and other nonlinear modeling problems by S. Chen *et al.* [10, 11]. This procedure is described in Appendix A.2.

Backward elimination (BE). In contrast to the previous methods, the backward elimination method starts building the regression (5.7) by using all wavelets in W, then eliminates one wavelet per stage, while trying to increase as less as possible the residual at each stage. This procedure is described in Appendix A.3.

5.5 Combining regressor selection and backpropagation

Any of the above procedures can be used to initialize wavelet network (5.4). This network is then further trained using a backpropagation procedure. Note that in (5.4) we use vectorial dilation parameters a_i, but for the regressor selection procedures the dilation parameters a_i in W are scalars. Before applying any backpropagation procedure, change the scalar dilation parameters resulting from the regressor selection procedures into vectors with identical components. Standard backpropagation is a stochastic gradient procedure, a quasi-Newton algorithm is however preferred for training the wavelet network, due to the good performance of the initialization procedures. Finally, in order to better capture linear properties in regressions, we replace (5.4) by

$$f_n(x) = \sum_{i=1}^{n} u_i \varphi(a_i \star (x - t_i)) + c^T x + b \tag{5.9}$$

with the additional parameters $c \in \mathbf{R}^d$, $b \in \mathbf{R}$. The initialization procedures are slightly modified accordingly.

Chapter 6

Fuzzy models : expressing prior knowledge in nonlinear nonparametric models

6.1 Fuzzy rules and prior knowledge in nonparametric models

We first begin by introducing fuzzy models such as typically used in fuzzy control [48]. Several presentations are possible, see for instance [85]. The presentation we give now is slightly heterodox, but is simple and consistent.

1. Input variables are scalar and are written $x_1, \ldots, x_d$. Input locations are encoded via fuzzy set membership functions, i.e., functions $\mu_A(x_i)$ with values in $[0, 1]$ where symbol A is just a label ; fuzzy set membership function μ_A is the mathematical meaning of "fuzzy set A". Thus, for each actual value of x_i, the statement " `x_i is A` " has a value equal to $\mu_A(x_i)$, such statements are premises of so-called "fuzzy rules". Be careful that a typical form of such statements is " `x_i is large` ", which does not convey as much information as formula $\mu_A(x_i)$ does, since function μ_A is not explicitly specified by this statement.

2. Fuzzy sets can be combined using the " `and, or, not` " operators of first order predicate logic. For instance,

   ```
   (x_1 is A_1) and (x_2 is A_2) ... and (x_d is A_d)
   ```

   is a fuzzy set involving the vector $(x_1, \ldots, x_d)$. Keyword "`and`" is a combinator of fuzzy sets which must be defined formally in terms of combination of membership set

functions. Several choices have been proposed by the various authors [32], the most widely used ones are

$$\begin{aligned} \mathtt{and}(u,v) = \min(u,v) \quad &, \quad \mathtt{or}(u,v) = \max(u,v) \\ \mathtt{and}(u,v) = uv \quad &, \quad \mathtt{or}(u,v) = u+v-uv \\ \mathtt{and}(u,v) = \max(0,u+v-1) \quad &, \quad \mathtt{or}(u,v) = \min(1,u+v) \end{aligned} \tag{6.1}$$

(corresponding definitions for **and** and **or** are written on the same line) and $\mathtt{not}(u) = 1-u$. Then, as usual in logic, implication " **(x is A) implies (y is B)** ", also written

```
if  x is A  then  y is B
```

is a macro which expands into [1]

```
(y is B)  or  not(x is A)
```

In the sequel, we shall encode the " **and***" as the product:* $\mathtt{and}(u,v) = uv$*, with corresponding codings for the "* **not**, **or***". Finally the implication is expanded as lastly stated.*

3. Fuzzy rules are statements of the form

```
if  x is A  then  y is B
```

Note that more complex premises can be used, using **and**, **or**, **not**. Here we state the mathematical translation of the classical "modus-ponens" mechanism, which writes

```
        rule   :  if  x is A   then  y is B
        fact   :      x is A'
-------------------------------------------
conclusion     :                  y is ?B
```

Modus-ponens is a mechanism which combines membership functions and yields a membership function, it can be viewed as a mechanism to express interpolation. Denote by $\mu_A(x)$ the membership function associated with fuzzy set **x is A** , and by $\mu_{A\Rightarrow B}(x,y)$ the membership function of " **if x is A then y is B** ". We now state the mathematical translation of the modus-ponens [32]. It is defined as :

$$\begin{aligned} \mu_{?B}(y) &= \mathrm{proj}_u \{\mu_{A'}(u) \ \mathtt{and} \ \mu_{A\Rightarrow B}(u,y)\} \\ &\triangleq \max_u \{\mu_{A'}(u) \ \mathtt{and} \ \mu_{A\Rightarrow B}(u,y)\} \end{aligned} \tag{6.2}$$

where elimination of component u has been performed via maximization. We now consider the particular case in which the fact is a *crisp* statement, i.e., has the standard

[1] This is the point where we deviate from the usual presentation: in the fuzzy litterature, implication is often encoded as a " **and** ", and the modus-ponens mechanism is modified accordingly. We preferred this presentation, since it is fully consistent and in accordance with the usual predicate calculus.

form " `x is` x" where x is an ordinary value. In this case, we have $\mu_{A'}(u) = 1$ if $u = x$, and $\mu_{A'}(u) = 0$ otherwise. Hence for such a case, the modus-ponens mechanism (6.2) reduces to

$$\mu_{?B}(y) \quad = \quad \mu_{A \Rightarrow B}(x, y) \;=\; 1 - \mu_A(x)\,(1 - \mu_B(y)) \tag{6.3}$$

where we have used the formulas $u \Rightarrow v = v$ `or not` $u = v + (1 - u) - v(1 - u) = 1 - u(1 - v)$. To conclude, since we only consider crisp facts, fuzzy rule

```
if  x is A  then  y is B
```

shall represent fuzzy set (6.3).

4. A "fuzzy rule basis" is a collection of fuzzy rules of the form, say,

```
if  (x_1 is A_1_1) and (x_2 is A_1_2) ... and (x_d is A_1_d)  then  (y is B_1)
...........
if  (x_1 is A_p_1) and (x_2 is A_p_2) ... and (x_d is A_p_d)  then  (y is B_p)
```

where the $A_{j,i}$ are doubly indexed labels, i is the index of the input coordinate, and j is the index of the rule. The mathematical translation of this rule basis is now given. We assume that the fuzzy sets form a *"fuzzy partition"* of the space, i.e.,

$$\sum_{j=1}^{p} \prod_{i=1}^{d} \mu_{A_{j,i}}(x_i) \quad \equiv \quad 1 \tag{6.4}$$

Then, *combining fuzzy rules within our fuzzy rule basis is interpreted as taking the "*`and`*" of their conclusions.* Thus, using notations of item 3 above, the above fuzzy rule basis represents the fuzzy set `?B` equal to

```
y is ?B_1  and  ...  and  y is ?B_p
```

where the `?B_j`'s are defined according to (6.3). Expressing the `and` combinator as the product of membership functions, we get

$$\begin{aligned}
\mu_{?B}(y) \quad &= \quad \prod_{j=1}^{p} \mu_{?B_j}(y) \\
\text{(by (6.3))} \quad &= \quad \prod_{j=1}^{p} \left(1 \;-\; \prod_{i=1}^{d} \mu_{A_{j,i}}(x_i)\,\big(1 - \mu_{B_j}(y)\big) \right) \\
&\approx \quad 1 \;-\; \sum_{j=1}^{p} \big(1 - \mu_{B_j}(y)\big) \prod_{i=1}^{d} \mu_{A_{j,i}}(x_i) \\
\text{(by (6.4))} \quad &= \quad \sum_{j=1}^{p} \mu_{B_j}(y) \prod_{i=1}^{d} \mu_{A_{j,i}}(x_i)
\end{aligned} \tag{6.5}$$

where we have used the property (6.4) of "fuzzy partition", and approximation $\prod_{j=1}^{p}(1-u_j) \approx 1-\sum_{j=1}^{p} u_j$, which is valid for u_j small and p large. Next, we also assume that sets B_j are *crisp,* i.e., they are of the form " `y is` y_j", thus $\mu_{B_j}(y) = 1$ if $y = y_j$, $= 0$ otherwise. Hence, assuming that both the consequences of the rules and the facts are crisp statements, we get for the conclusion the fuzzy set " `y is ?B` ", where

$$\mu_{?B}(y) \quad = \quad \sum_{j=1}^{p} 1_{\{y=y_j\}} \prod_{i=1}^{d} \mu_{A_{j,i}}(x_i) \ . \tag{6.6}$$

At this point, setting $x = (x_1, \ldots, x_d)$, formula (6.6) defines a function mapping points $x \in \mathbf{R}^d$ into fuzzy sets. To get a function in the usual setting $\mathbf{R}^d \mapsto \mathbf{R}$, we perform *defuzzification* of $\mu_{?B}(y)$ in (6.6), i.e., we replace $\mu_{?B}$ by its center of gravity, using again fuzzy partition property (6.4), see [48] [32]. This finally yields the ordinary function

$$y \quad = \quad \sum_{j=1}^{p} y_j \left(\prod_{i=1}^{d} \mu_{A_{j,i}}(x_i) \right) \triangleq \sum_{j=1}^{p} y_j \ w_j(x) \ , \tag{6.7}$$

where $x = (x_1, \ldots, x_d)$, this defines the weights $w_j(x)$. If property (6.4) does not hold, i.e., if our fuzzy rule basis is sparse so that the range of each coordinate x_i is not covered by a fuzzy partition, then the above defuzzification formula is modified accordingly :

$$y \quad = \quad \frac{\sum_{j=1}^{p} y_j \ w_j(x)}{\sum_{j=1}^{p} w_j(x)} \ . \tag{6.8}$$

Usually, fuzzy set membership functions are parametrized functions of the form

$$\mu_A(x) \quad = \quad \mu(a(x-t)) \tag{6.9}$$

where $\mu(x)$ is a given function with values in $[0,1]$, a is a dilation factor, and t is a translation factor, and the pair (a,t) encodes the fuzzy set A. Mostly used is the piecewise linear function μ such that $\mu(1) = 1$ and $\mu(x) = 0$ for x outside the interval $[0,2]$, i.e., a spline of order 1. In this case, the defuzzification mechanism (6.7) just performs interpolation. If the fuzzy partition is fixed and not adjustable, then we get a particular case of the Kernel estimate (2.1). Obviously, fuzzy models such as (6.7) or (6.8) are amenable of identification since they have some unknown parameters for tuning, namely the y's, a's, and t's. Identified fuzzy models are often referred to as "neuro-fuzzy models" in the A.I. litterature [39], since standard backpropagation (i.e., stochastic gradient) can be used for their training, exactly as for neural networks. It is also proved that fuzzy models are universal approximants [81], which is not surprising.

To summarize, fuzzy models are described by fuzzy rule bases, plus some additional parameters which make vague statements such as "large", "small", etc., to be precise in terms of fuzzy set membership functions. The fuzzy rule basis exhibits the structure of the model, plus some coarse features related to the location of the elementary functions in the

decomposition (6.7) or (6.8). Thus *fuzzy models are just particular instances of the kind of nonlinear nonparametric model we consider here, with the advantage of providing the fuzzy rules as a way to describe some possibly available prior knowledge.* In the experiments reported in section 7.1.2, neuro-fuzzy modelling is used in the above sense.

6.2 Fuzzy rule bases for wavelet based estimators

In this short section, we briefly discuss a proposal for blending the practical advantages of fuzzy models with the mathematical quality of wavelet based identification techniques. Further development of this proposal will be the subject of future work and will be reported elsewhere.

Requirements

Formulae (6.4) and (6.7) reveal that fuzzy models can be viewed as interpolation procedures : interpolation is performed between points where the set membership function takes value 1, with associated y value. Thus fuzzy models cannot reflect hierarchical or multiresolution approximations of a function such as performed by wavelet based identification techniques. So the following natural question can be considered : *how to provide fuzzy rule bases for wavelet based estimators*? Thus what we need is to abstract wavelet networks, say, of the form (5.4), in the form of syntax similar to fuzzy rule bases. Such syntax would not specify the considered wavelet network exactly, but should capture some essential features of it. Objectives would be to use such a syntax for a rough but easy description of a wavelet network based on some qualitative prior knowledge on the system, or to use it as an initial guess for some iterative identification procedure based on recorded data from the system.

Reflecting the notion of multiresolution or hierarchy within rules calls for a synctactic notion of *context.* For instance, in the context " `x is large` ", we may want to write " `x is small` " to express that `x` is not too large, and " `x is large` " again to insist that `x` is very large indeed. This calls for logics handling context dependent statements. Such logics are studied under different frameworks independently in the A.I. and theoretical computer science communities. The notion of a "conditional object" proposed and studied by Didier Dubois and Henri Prade [31] in the A.I. community is a candidate model for such "context dependent rules". In [31] various definitions are investigated for such "conditional objects", based on some reasonable requirements accepted as axioms. On the other hand, "structured operational semantics" (SOS) was introduced by Gordon Plotkin [64] in theoretical computer science. SOS rules describe the legal transitions of a considered program *for a given context.* SOS rules are used to specify primitives as well as the various combinators for program contruction. We shall not elaborate any further on possible theoretical models for the kind of context dependent statements we shall take the liberty to write in the sequel.

Let us propose the following syntax we call *hierarchical fuzzy models.*

1. Standard fuzzy rules are hierarchical fuzzy rules. Thus we can still write

```
if  (x_1 is A_1) and (x_2 is A_2) ... and (x_d is A_d)  then  (y is B)
```

with the same mathematical meaning as before.

2. Let us give names to fuzzy rule bases, e.g.,

```
RULE_BASE is

if  (x_1 is A_1_1) and (x_2 is A_1_2) ... and (x_d is A_1_d)
    then  (y is B_1)
.....
if  (x_1 is A_p_1) and (x_2 is A_p_2) ... and (x_d is A_p_d)
    then  (y is B_p)

end
```

Then the following statement

```
if  (x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)  then RULE_BASE applies
```

is a *hierarchical fuzzy rule.* Its premise is an ordinary fuzzy statement

```
(x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)
```

as before. The second part of this statement, namely " `then RULE_BASE applies` " has " `then` " and " `applies` " as keywords and `RULE_BASE` as a parameter. This hierarchical fuzzy rule has the following interpretation :

(a) The reference space for input x, which was, say, $[0,1]^d$, is now stretched down, to enforce validity of the statement

```
(x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)
```

Thus all premises of `RULE_BASE` are stretched down accordingly.

(b) Since `RULE_BASE` was a standard fuzzy rule basis, our new rule is a hierarchical fuzzy rule.

3. Collections of hierarchical fuzzy rules are termed *hierarchical fuzzy rule bases.* Hierarchical fuzzy rules can call hierarchical fuzzy rule bases, this captures multiresolution. Obviously, in doing so, the question of *recursivity* in the computer science setting occurs : does it happen that a rule recursively calls itself? Recursion may or may not be accepted. Anyway, simple syntactic constraints in writing rule bases would prevent from recursivity.

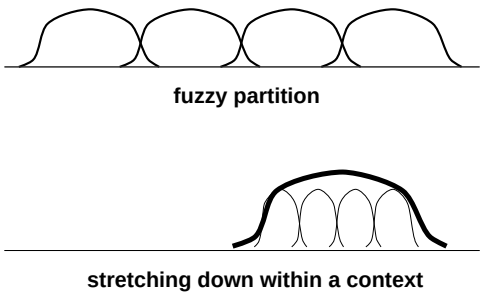

Figure 6.1: *The down stretching mechanism.*

The "down stretching" mechanism

The key issue in this informal discussion is the precise mathematical meaning of the "down stretching" mechanism. We assume for convenience that the default context is $[0,1]^d$. Consider a fuzzy partition satisfying condition (6.4), which we recall now

$$\sum_{j=1}^{p} \prod_{i=1}^{d} \mu_{A_j}(x_i) \quad \equiv \quad 1 \tag{6.10}$$

Down stretching this fuzzy partition to a given membership function $\mu_C(x_i)$, consists in building a collection $\mu_{(A_{j,i}|C)}, j = 1, \ldots, p$ of membership functions which satisfy

$$\sum_{j=1}^{p} \prod_{i=1}^{d} \mu_{(A_{j,i}|C)}(x_i) \quad \equiv \quad \mu_C(x_i) \tag{6.11}$$

and, in addition, preserve the "geometry" of the original fuzzy partition. This is illustrated in Figure 6.1. A possible procedure achieving this is described now.

We first need to define the notion of a fuzzy set more accurately. A fuzzy set A is a triple $A = (\mu_A(x), a, b)$, where

$$\mu_A : [a,b] \rightarrow [0,1] \text{ is the membership function, and}$$
$$-\infty < a \leq b < +\infty \ .$$

The interval $[a,b]$ is the context of the fuzzy set A. For example, when we define a fuzzy set " `small` ", we must specify its context interval $[a,b]$ in addition to its membership function. This " `small` " label means that the μ_A membership function is mainly concentrated on the small values of this context interval. Note that this set may not be " `small` " within other context intervals.

Now consider a fuzzy set $C = (\mu_C(x), a', b')$, we consider its left and right boundaries defined by :

$$l_C = \inf_{\mu_C(x)>0} x \quad , \quad r_C = \sup_{\mu_C(x)>0} x \ ,$$

i.e., $[l_C, r_C]$ is the support of μ_C. Consider a pair (A, C) of fuzzy sets, define the *contextual fuzzy set* $(A|C)$ as follows :

$$\begin{aligned} (A|C) &= (\mu_{(A|C)}, l_C, r_C) \\ \mu_{(A|C)}(x) &= \mu_A\left(\frac{b-a}{r_C - l_C}(x - l_C) + a\right)\mu_C(x) \ . \end{aligned} \tag{6.12}$$

Hence $(A|C)$ has the support of C as context, and its membership function is obtained by mapping the interval $[a, b]$ onto $[l_C, r_C]$, and then by multiplying by μ_C. With this definition of contextual fuzzy sets, a fuzzy partition having the default context, i.e., satisfying property (6.10), is down stretched to a fuzzy partition satisfying property (6.11).

Mathematical implementation of the hierarchy

Here we formalize what it means for a rule base to be called within a given context. As an example, we give the meaning of the hierarchical statement

```
if  (x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)  then (y = y_o)
if  (x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)  then RULE_BASE applies
```

where `RULE_BASE` has been defined before. We may also rewrite this as

```
if  (x_1 is C_1) and (x_2 is C_2) ... and (x_d is C_d)  then (y = y_o)
                                                        and RULE_BASE applies
```

First, we have to combine two rules, and this is performed using the general formula (6.7). Then we must recall that `RULE_BASE` is called within the context of $C_1 \times \ldots \times C_d$ hence we use definition (6.12) of contextual fuzzy sets. This yields the following mathematical interpretation of the above hierarchical rule base :

$$y = y_o\left(\prod_{i=1}^{d} \mu_{C_i}(x_i)\right) + \left[\sum_{j=1}^{p} y_j \left(\prod_{i=1}^{d} \mu_{(A_{j,i}|C_i)}(x_i)\right)\right] \tag{6.13}$$

This shows that the value y_o can be interpreted as a "first order approximant", while the y_j's, $j = 1, \ldots, p$, are increments corresponding to a refinement of our modelled function. Thus *truncating such an approximation is simply performed by truncating the tree of the nested calls of rule bases.*

Thus what we have at this point is a flexible way to associate syntax with multiresolution expansions of functions. If, in addition, *we carefully choose our membership functions*

μ_A to be derived from scale functions φ associated with wavelets, we now have a way to abstract wavelet networks in the form of hierarchical fuzzy rule bases. See Section 3.1 for scale functions which are nonnegative and bounded, and thus satisfy the requirements for being prototypes of membership functions. The "call" mechanism provides some kind of genericity, since the same rule base can be called within different contexts. This genericity is expected to be useful mainly when adjustable parameters, which are hidden inside fuzzy rules, are identified from data. On the other hand, for fuzzy models specified based on the prior knowledge of the user, it is not expected that the same rule base will be called under different contexts.

Chapter 7

Experimental results

In this chapter we consider the application examples introduced in chapter 1.1. We provide detailed results obtained with the wavelet networks and the fuzzy network. For the gas turbine example, we also compare them with alternative semi-physical models which were developed in [3] for the purpose of monitoring and diagnostics.

7.1 Modelling the gas turbine system

7.1.1 Using the wavelet network

In the gas turbine system we introduced in Section 1.1.1, the temperature profile at the exhaust of the turbine is considered as the output. We need a model which predicts this temperature profile from available measurements. For the semi-physical model we mentioned in subsection 1.1.1, the temperature profile is predicted from the mean temperature in the combustion chambers T_e, the mean temperature at the exhaust of the turbine T_s, and the rotation velocity of the turbine N. Velocity N is directly measured, T_s is given by the average of a set of thermocouples installed at the exhaust of the turbine, T_e is computed from T_s and the compression rate π of the compressor [86, 90]. By substituting T_e, the temperature profile at the exhaust of the turbine depends on T_s, π and N. As suggested by this semi-physical model, we assume that the temperature measured by each of the thermocouples installed at the exhaust of the turbine is a function of T_s, π and N, which all are measured. Therefore, we can try to construct, for each of the thermocouples, a wavelet network with T_s, π and N as its input variables, and train it to predict the temperature measured by the thermocouples.

We have experimented this approach on the data taken from a gas turbine of European Gas Turbine SA. The training data were collected during about 48 hours. We have resampled the data and kept only 1000 measurement points. This gas turbine system is equipped with 18 thermocoupes at its exhaust. For the sake of brevity, we show only the results concerning the

first thermocouple. The resampled data are depicted in figure 7.1 where the plots correspond to T_s, π, N and $y = t_1 - T_s$, where t_1 is the measurement of the first thermocouple. These 1000 measurement points, which we refer to as the *training data*, are used for training models whose input vector is $x = (T_s, \pi, N)^T$. The obtained models are tested on another set of measured data, which we refer to as the *test data* set and depict in figure 7.2.

We have chosen the radial wavelet function $\varphi(x) = (d - x^T x)e^{-\frac{1}{2}x^T x}$ with $d = \dim(x)$. The number of wavelets used in the networks is set to 40. Note that there are 18 thermocouples.

We initialize the wavelet networks with each of the proposed (RBS, SSO, BE) procedures and train them with the Gauss-Newton procedure.

In order to show the performance of the resulting models, we compare their results with those of the semi-physical model and a third order polynomial model. In figures 7.3 and 7.4 the results obtained with the semi-physical model and the third order polynomial model, are respectively shown. The results obtained with the wavelet networks initialized with procedures RBS, SSO, BE, and the results after 10 iterations of the Gauss-Newton procedure, are given in figures 7.5, 7.6 and 7.7. In table 7.1 we listed the mean of square errors (MSE) of these models on the training data set as well as on the test data set. For each of these networks we give the result of its initialization (init. MSE) and the result after 10 iterations of the Gauss-Newton procedure (final MSE). The time of computation for building these models is also listed in table 7.1, based on our programs in MATLAB 4.1 language executed on a Sun Sparc-2 workstation. Since the execution time of the programs is perturbed by other processes on the workstation, another figure of merit is provided, namely the the MATLAB's Flop which measures the computational complexity of a program.

The following observations can be made :

- The semi-physical model performs quite poorly in predicting the output of the system.
- The system is truly nonlinear, in addition the results obtained with the polynomial model are quite poor.
- The wavelet networks do improve the performance on prediction. But recall we get in turn increasing computational complexity and loss of the physical meaning of the model parameters.

7.1.2 Using the fuzzy network

We also applied the (classical) neuro-fuzzy network as briefly introduced in Section 6 for modelling the gas turbine system. Similarly to the wavelet network, we train the fuzzy network using the training data set, and then evaluate it on the test data set.

To build the network, we have taken a fuzzy partition of the state space using triangular membership functions (i.e., first order splines), this divides the variation domain of each input into five equal parts. Following Section 6, the mathematical translation for both conjunction and implication operators is taken to be the product.

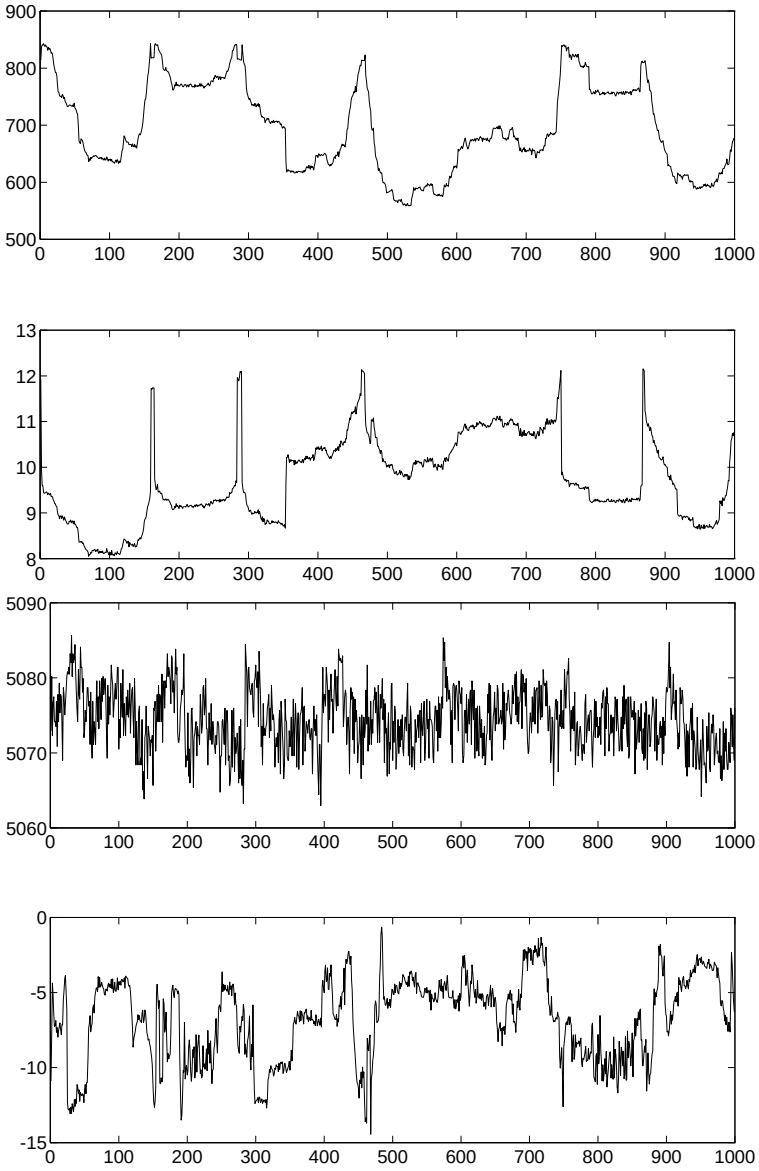

Figure 7.1: Training data. The plots correspond to, from top to bottom, T_s, π, N and $y = t_1 - T_s$.

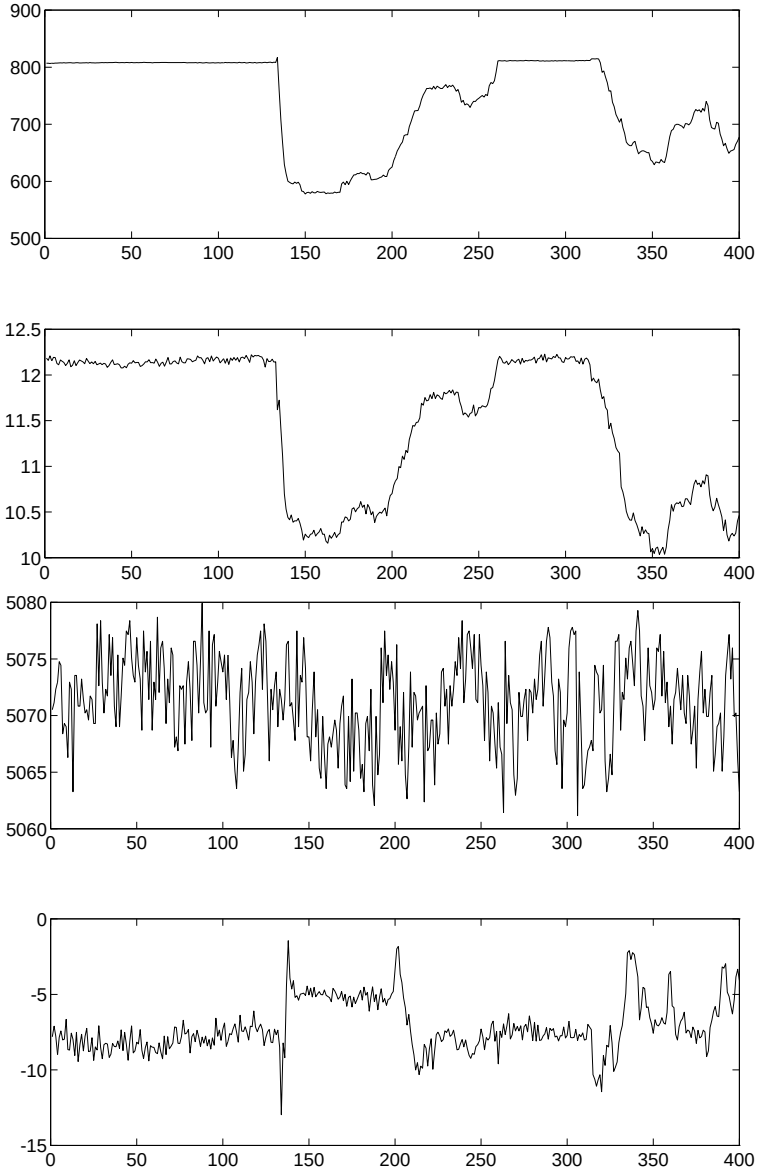

Figure 7.2: Test data. The plots correspond to, from top to bottom, T_s, π, N and $y = t_1 - T_s$.

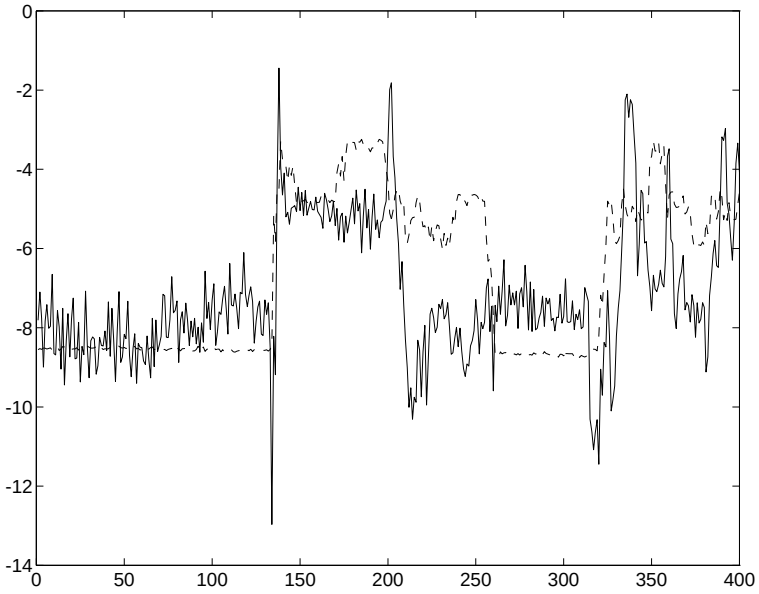

Figure 7.3: Result with the semi-physical model on the test data set. The solid line represents the true measurement and the dashed line represents the output of the model.

models	RBS-net	SSO-net	BE-net	semi-physical	polynomial
train. init. MSE	1.2656	1.0453	1.0381		
train. final MSE	0.5395	0.4239	0.4503	3.5268	2.8438
test. init. MSE	1.2368	1.1229	1.1576		
test. final MSE	1.1886	1.2348	1.0898	2.8914	2.1135
init. flops	2.0718×10^7	4.3714×10^8	7.5143×10^7		
train. flops	1.5365×10^9	1.5365×10^9	1.5365×10^9	9.8041×10^8	4.7056×10^5
init. time (sec.)	41.6	251.2	87.2		
train. time (sec.)	2461.8	2383.8	2456.5	2265.0	1.5362

Table 7.1: Performance evaluation of the models

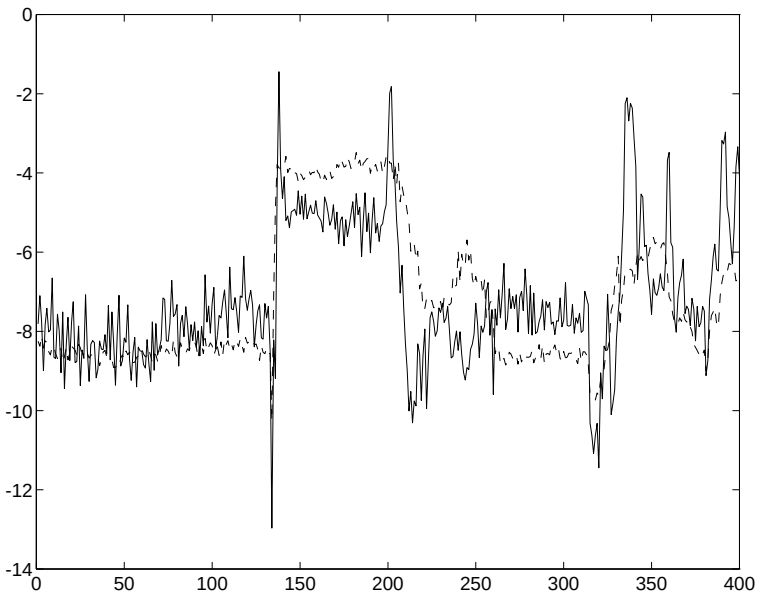

Figure 7.4: Result with the third order polynomial model on the test data set. The solid line represents the true measurement and the dashed line represents the output of the model.

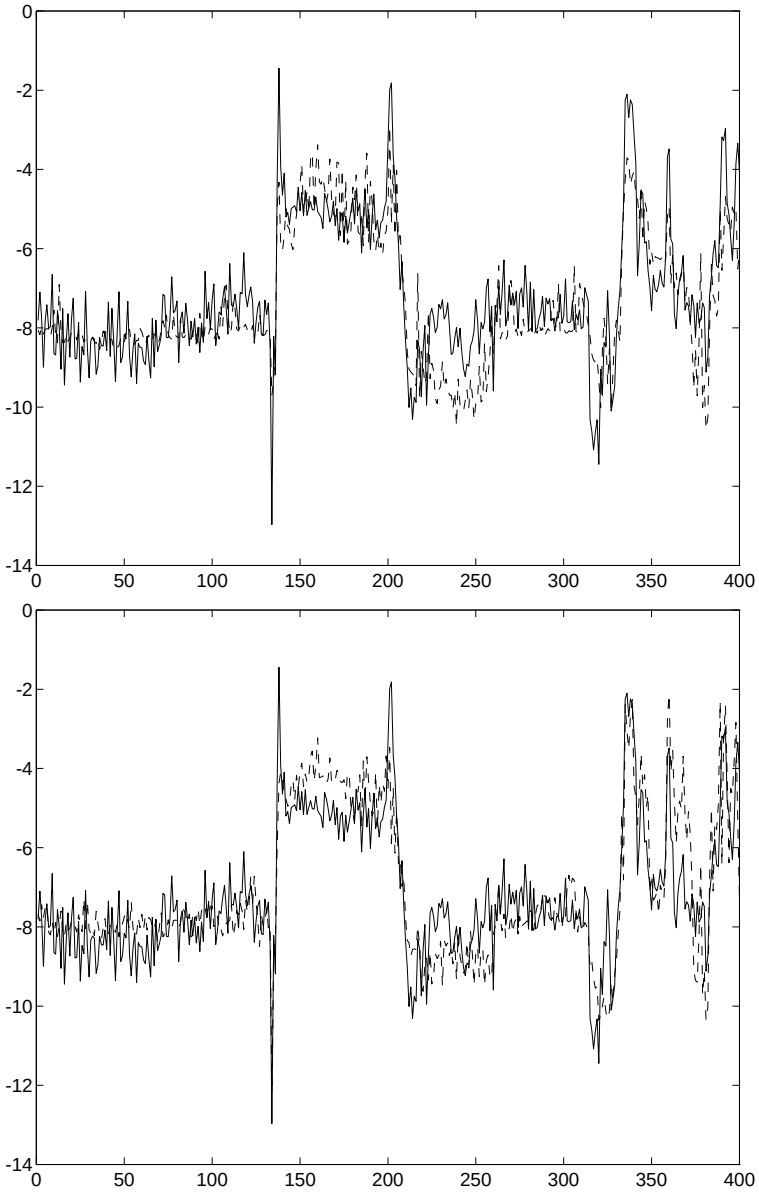

Figure 7.5: Results with wavelet network initialized by procedure RBS (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

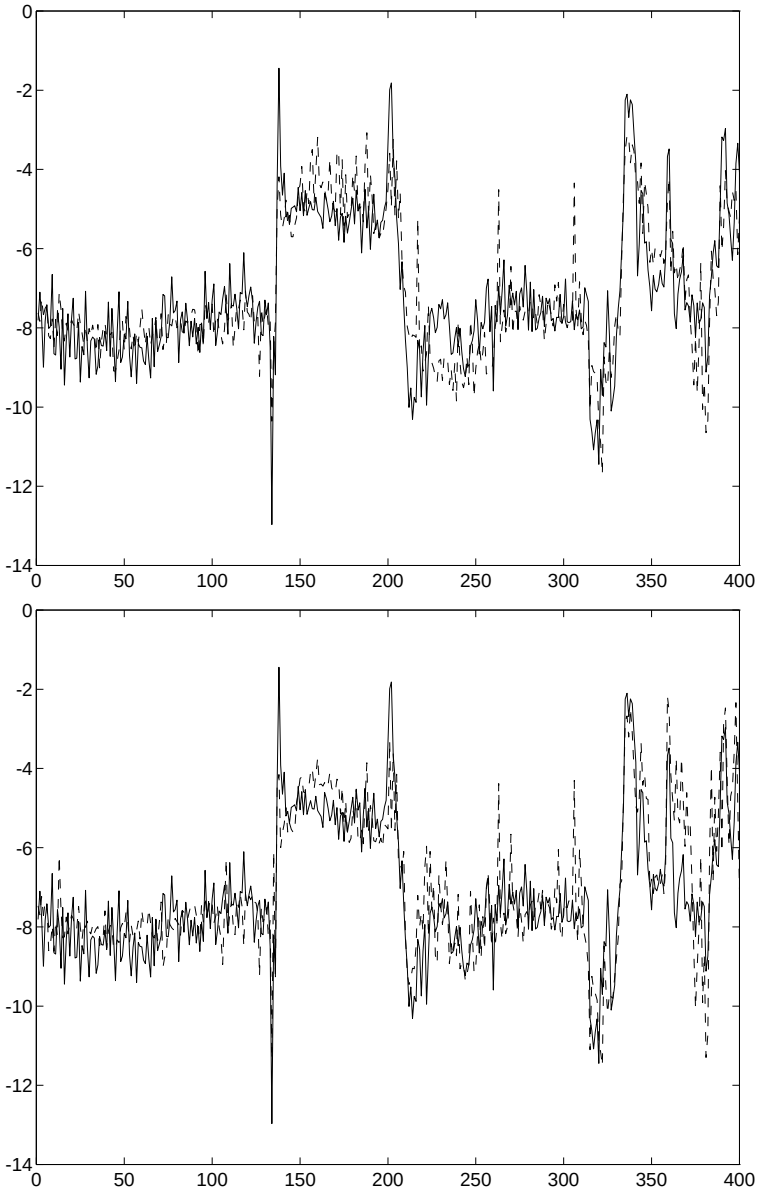

Figure 7.6: Results with wavelet network initialized by procedure SSO (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

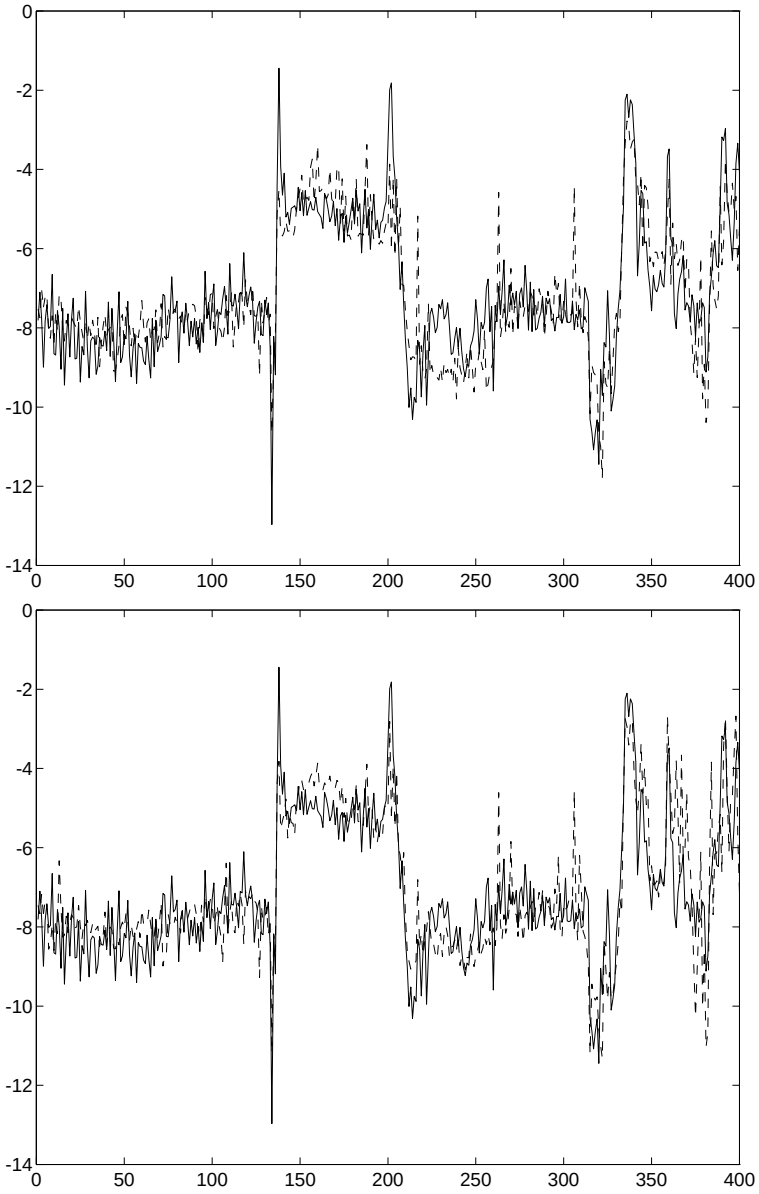

Figure 7.7: Results with wavelet network initialized by procedure BE (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

Before learning, we have initialized the network using a simple interpolation procedure. Consider "defuzzification" formula (6.7) which we recall now:

$$y \;=\; \sum_{j=1}^{p} y_j \;\left(\prod_{i=1}^{d} \mu_{A_{j,i}}(x_i)\right) \;\triangleq\; \sum_{j=1}^{p} y_j \; w_j(x) \;, \tag{7.1}$$

where index j labels the rules. For each rule j, select the training input data point X_{n_j} closest to the center of the corresponding fuzzy set, i.e., $w_j(x)$ is maximal for $x = X_{n_j}$. Then take $y_j = Y_{n_j}$ where Y_{n_j} is the output value corresponding to X_{n_j}. Results of this procedure are shown in Figure 7.8.

The second stage consists in performing a least squares fit of the parameters θ_j in function $f_\theta(x) = \sum_{j=1}^{p} \theta_j \; w_j(x)$, where $\theta = (\theta_1, \ldots, \theta_p)$ based on the whole training data sample $\mathcal{O}_1^N = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$. A bruteforce implementation of least squares would be difficult, due to the need for inverting the Hessian of the least squares functional. Thus an iterative stochastic gradient procedure has been preferred instead, using the above simple initialization technique. Training was stopped after only three successive scanning of the learning set.

The identified fuzzy network is then evaluated on the test data set. The output of the identified fuzzy network is plotted in Figure 7.8, and is compared to the actual one. The solid line represents the true measurement and the dashed line represents the output of the model. The mean of square errors (MSE) on the test data set is 1.5860.

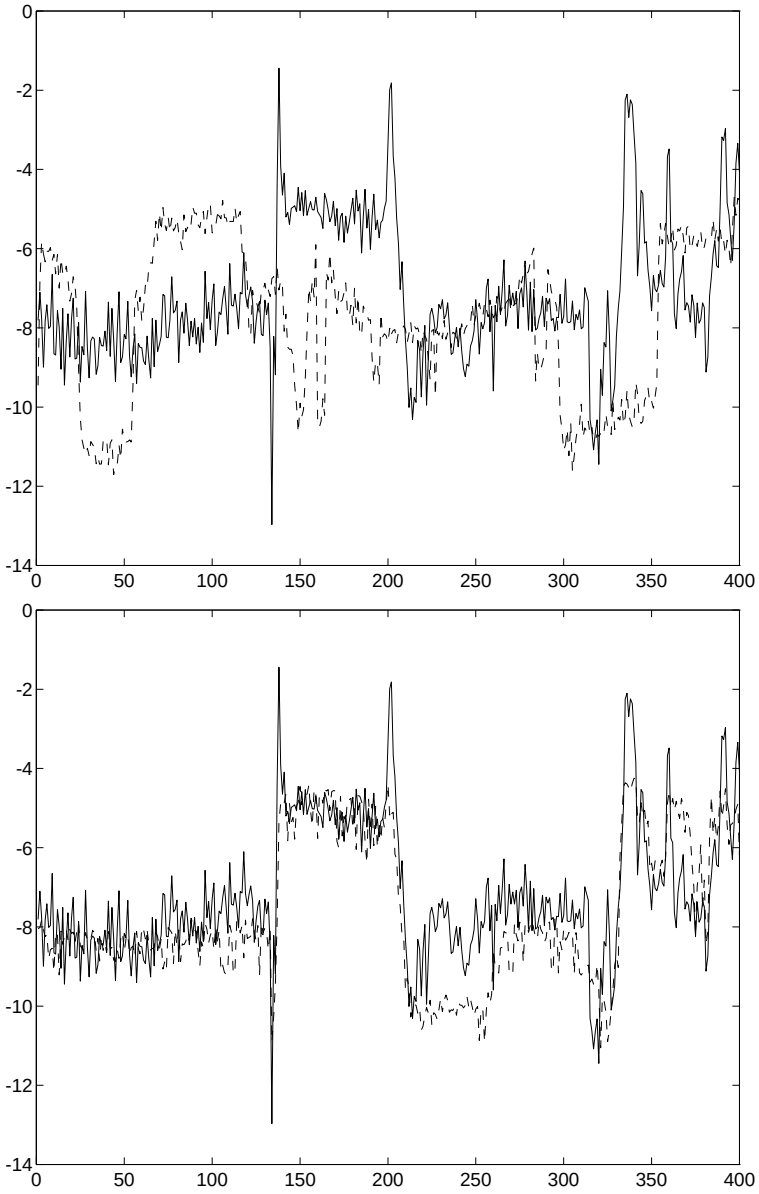

Figure 7.8: *Results with the initialized (top) and trained (bottom) neuro-fuzzy networks on the test data set. The solid lines represent the true measurement and the dashed lines represent the output of the model.*

7.2 Modelling the hydraulic actuator of the robot arm

Let us denote by $u(t)$ and $p(t)$ the position of the valve and the oil pressure at time t, respectively. A sample of 1024 pairs of $(u(t), p(t))$ was registered[1]. We divide it into two equal parts for training and testing the models. The training data are depicted in figure 7.9, and the test data in figure 7.10.

We first tried to model this system with linear autoregressive exogenous (ARX) models. More precisely, we tried to use models of the following form:

$$\begin{aligned} p(t) &= a_1 p(t-1) + a_2 p(t-2) + \cdots + a_n p(t-n) \\ &\quad + b_1 u(t-\tau-1) + b_2 u(t-\tau-2) + \cdots + b_m u(t-\tau-m) + e(t) \end{aligned}$$

where the pure time delay τ is assumed to be an integer and $e(t)$ is some noise independent of $u(t)$ and past values of $p(t)$. After the identification of the model parameters a_i, b_j, τ, we plot the output of the following system to visually evaluate the quality of the model:

$$\begin{aligned} \widehat{p}(t) &= a_1 \widehat{p}(t-1) + a_2 \widehat{p}(t-2) + \cdots + a_n \widehat{p}(t-n) \\ &\quad + b_1 u(t-\tau-1) + b_2 u(t-\tau-2) + \cdots + b_m u(t-\tau-m) \ . \end{aligned}$$

We processed the data with L. Ljung's System Identification Toolbox, Version 3.0a. It turns out that the ARX model that gives the best simulation result on the test data set has the model order with $n = 3, m = 2, \tau = 0$. This result is shown in figure 7.11. It does not seem to be satisfactory. The wavelet networks as defined in (5.9) are then considered as candidates of nonlinear models.

In analogy with the linear ARX model, we build models of the following form:

$$p(t) = \widehat{f}(p(t-1), p(t-2), p(t-3), u(t-1), u(t-2)) + e(t)$$

where the nonlinear estimator $\widehat{f}$ is a wavelet network composed of 6 wavelets, and $e(t)$ represents the modelling error. To train the network, compose its input and output vectors with the training data $\{u(t), p(t)\}$:

$$\begin{aligned} x(t) &= [p(t-1), p(t-2), p(t-3), u(t-1), u(t-2)]^T \ , \\ y(t) &= p(t) \ . \end{aligned}$$

Then apply the initialization algorithms and the Gauss-Newton procedure. Again we take

$$\varphi(x) = (d - x^T x) e^{-\frac{1}{2} x^T x}$$

with $d = \dim(x)$ as the wavelet function. It happens that for this example, the Gauss-Newton procedure does not significantly improve the performance of the wavelet models, so we only show the results obtained with the initialized networks.

We then simulate the output $\widehat{p}(t)$ on the test data set with the wavelet models, in a similar way as with the linear ARX model:

[1] We gratefully acknowledge Jonas Sjöberg and Svante Gunnarsson from Linköping University for providing the data.

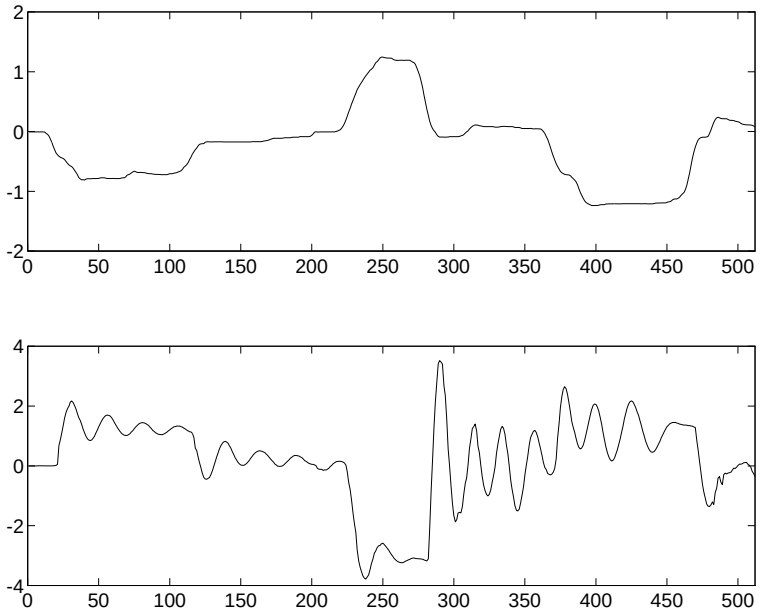

Figure 7.9: Training data: the input $u(t)$ (top) and the output $p(t)$ (bottom).

$$\widehat{p}(t) = \widehat{f}(\widehat{p}(t-1), \widehat{p}(t-2), \widehat{p}(t-3), u(t-1), u(t-2))$$

The simulation results obtained with the wavelet networks initialized with algorithms RBS, SSO and BE are depicted in figure 7.12 7.13 and 7.14.

Clearly, the wavelet models significantly improve the result of the simulation. While the results obtained with initialization algorithms SSO and BE are very similar, the result of algorithm RBS is obviously less good.

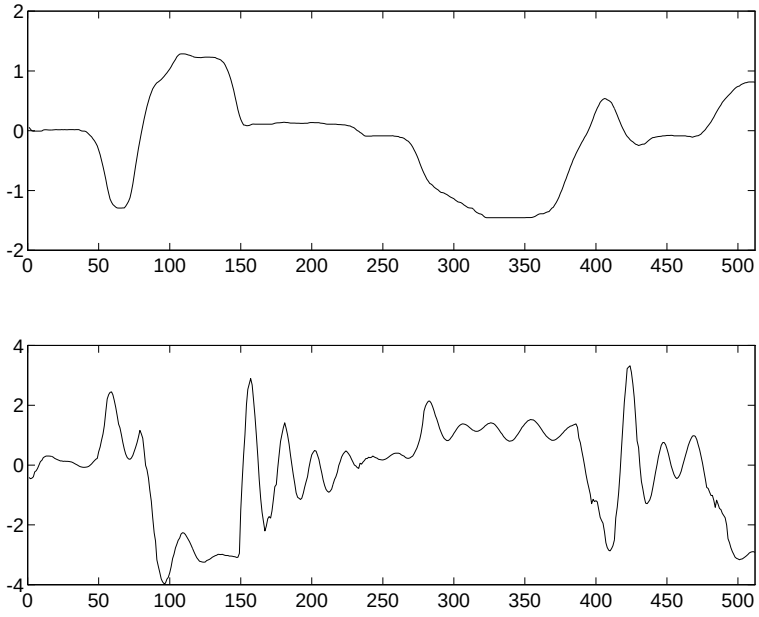

Figure 7.10: Test data: the input $u(t)$ (top) and the output $p(t)$ (bottom).

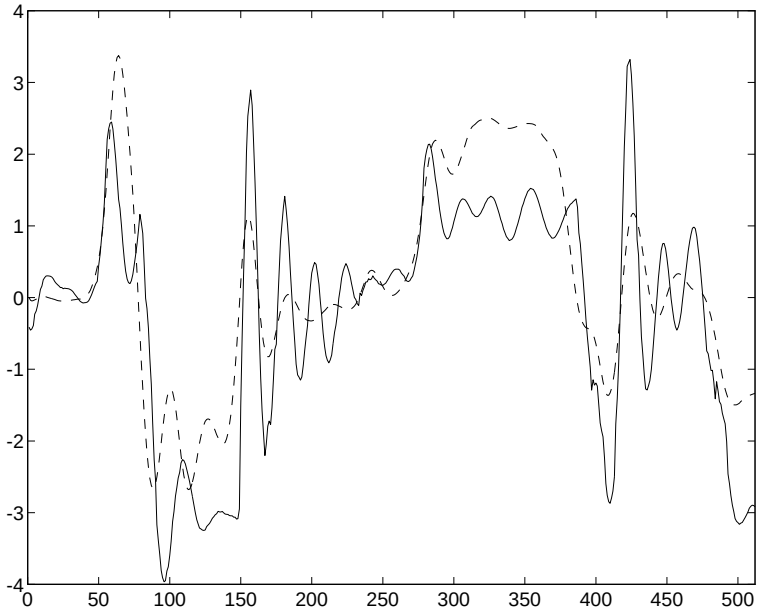

Figure 7.11: Result with the linear ARX model on the test data set. The solid line represents the true measurement and the dashed line represents the simulated output.

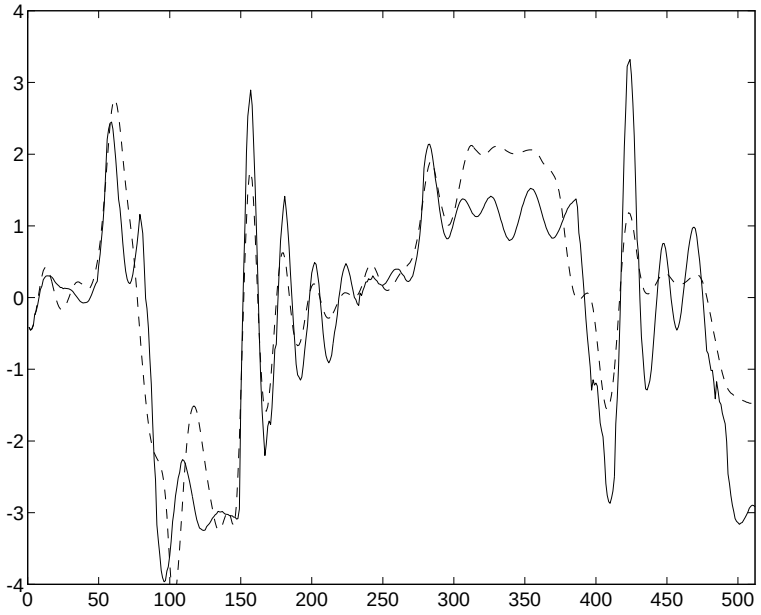

Figure 7.12: Result with the wavelet network initialized with algorithm RBS. The solid line represents the true measurement and the dashed line represents the simulated output.

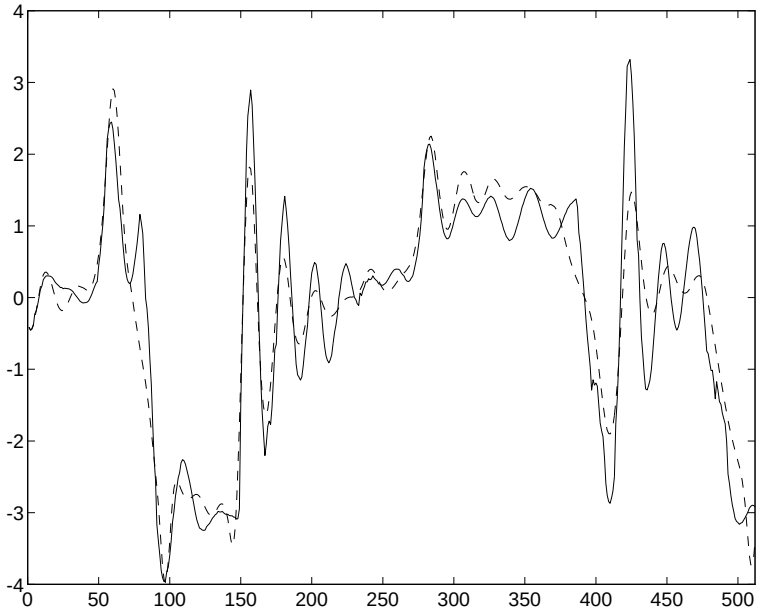

Figure 7.13: Result with the wavelet network initialized with algorithm SSO. The solid line represents the true measurement and the dashed line represents the simulated output.

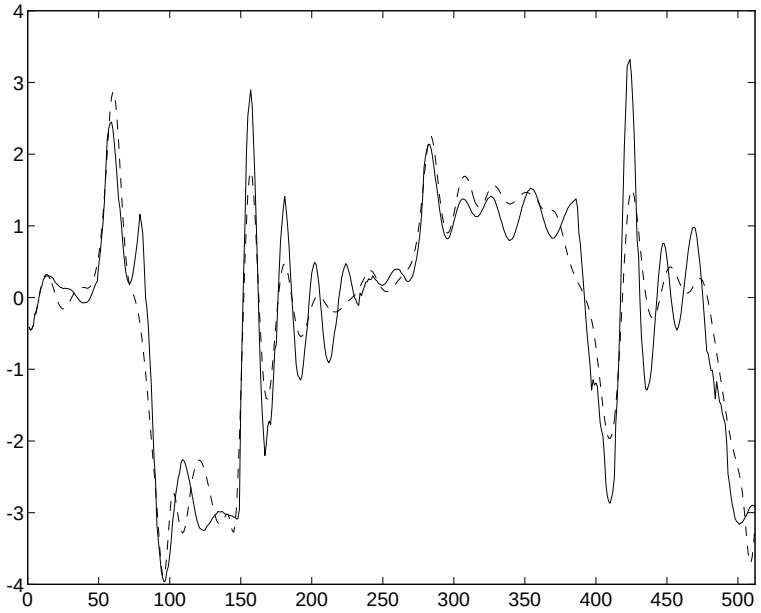

Figure 7.14: Result with the wavelet network initialized with algorithm BE. The solid line represents the true measurement and the dashed line represents the simulated output.

7.3 Predictive fuzzy modelling of glycæmic variations

7.3.1 The variables of interest and their qualitative labels.

Diabetologists' knowledge is expressed under the form of "rule of thumb" advices. We have used this knowledge to build a two hour ahead predictive model of glycæmic variations. This predictive model will be subsequently used in a control system. We have restricted our model to six inputs (current instant t is omitted for simplicity) :

item	symbol	fuzzy values
glycæmia	`Gl`	`Very Low` (`VL`) `Low` (`L`) `Normal` (`N`) `High` (`H`) `Very High` (`VH`)
basis insulin injection rate	`Ba`	`Low`, `Normal`, `High`
flash insulin injection rate	`Bo`	`Low`, `Normal`, `High`
elapsed time since previous meal	`Dr`	`Far Before`, `Near`, `Just After`, `Far`
diet	`Nr`	`Fiber` , `Normal`, `Glucidic`
expected future activity	`Ac`	`Low`, `Normal`, `High`

The ouput is the predicted variation of glycemia at time $t+2$ hours, `DG(t+2)` $\in$ {`PVB, PB, PM, PS, Z, NS, NM, NB, NVB`}, where `P` means "Positive", `N` "Negative", `S` "Small", `B` "Big", etc. Figure 7.15 shows membership functions of glycemia, where the $(g_i)_{i=0}^4$ parameters must be determined by learning since their optimal value depends on the patient. Membership functions have been represented by simple first order splines with free knots. Our method follows the following two steps :

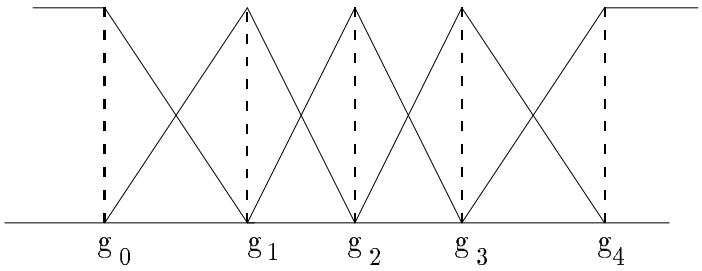

Figure 7.15: Fuzzy partition for glycemia

1. start with an initial guess of the model, based on available (qualitative) prior knowledge ;

2. tune this model to the particular patient in consideration, by performing learning from data.

7.3.2 Expressing prior knowledge

Combining all possible qualitative values for the different inputs yields 1620 different cases, corresponding to the same amount of candidate fuzzy rules. In fact, only 50 rules were considered for our prior model, thus reflecting the actual domain for the input variables where meaningful knowledge exists. Example of such rules are

```
if  (GL(t) is VL)  and  (Nr(t) is N)  then   DG(t+2) is PB
if  (GL(t) is  L)  and  (Ba(t) is L)  then   DG(t+2) is NS
```

Figure 7.16 shows predicted glycemia at $t+\delta$ from glycemia at time t, with $\delta = 2$ hour, *before learning,* i.e., with only use of the prior model. The solid line shows the actual glycæmia and the dashed line the predicted one. The doctor's rules are quite efficient in predicting the effect of insulin injections. Still some spikes occur in the prediction error. Prediction error has mean $\mu = -0.20$ and standard deviation $\sigma = 0.38$.

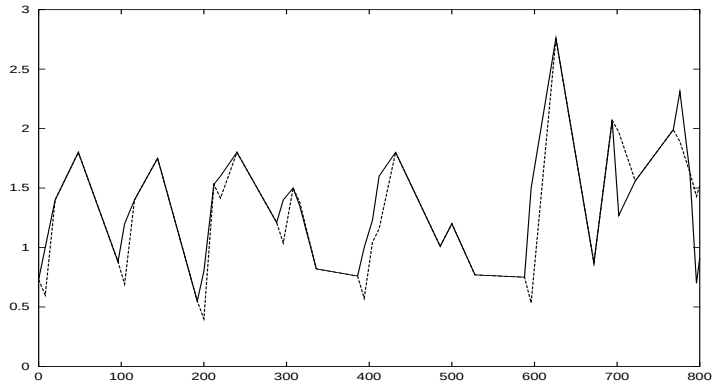

Figure 7.16: *Prior model:* two hour ahead prediction (dashed line) vs. actual (solid line) glycæmia

7.3.3 Tuning the model for each patient

Using data from patient's note-book, we divided data file into two parts, one for learning and the other for generalization (i.e., testing). Figure 7.17 shows predicted glycemia at $t+2$ from glycemia at time t, *after learning,* i.e., subsequent learning of the g_i parameters on data. A simple stochastic gradient was used. Prediction error has mean $\mu = -0.0003$ and standard deviation $\sigma = 0.29$. Some improvement is seen, note that such an improvement is likely to be patient dependent. The errors around time steps 700 and 800 are due to catheter changes (as marked in the note-book) which usually lead to inject more insulin than expected.

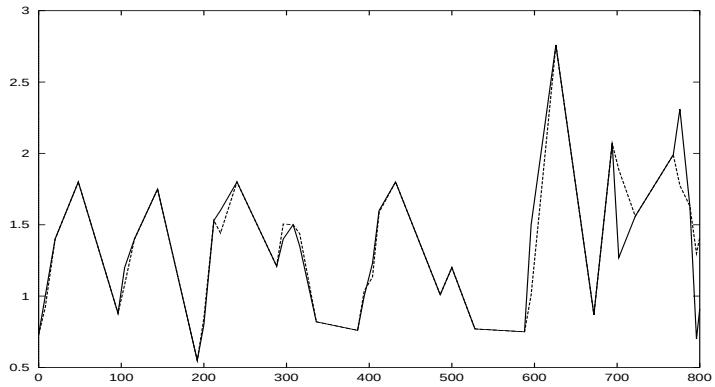

Figure 7.17: *Model after learning:* two hour ahead prediction (dashed line) vs. actual (solid line) glycæmia

7.3.4 Comments and conclusions about this example

The following conclusions can be drawn from this case study :

- Fuzzy rules turned out to be a convenient way to express prior knowledge from doctors, in part because this prior knowledge is mainly qualitative. It is important to notice that this fuzzy rule basis was far from being equivalent to an exhaustive table describing the input-output map, since about only a few percent (50/1620) of this table was described by the rules. This restriction is by itself a useful prior information about the range of validity of the modelling.

- Subsequent tuning of the prior model was performed while preserving the structure of the model, i.e., the fuzzy rules were not modified, only the g_i parameters hidden in the splines were adjusted. It would also be possible to use our prior model as initial guess but allow other "rules" (i.e., additional splines) to be introduced via learning ; corresponding experiments are under progress.

- Another advantage of describing the model via fuzzy rules is the possibility to "decompile" the model after learning, again in the form of fuzzy rules, for return to the user (doctor or patient). Returning a mathematical model would be of little use for the average user, having no training in mathematics.

- In this application, high accuracy was not a key point. For other cases where model accuracy is more important, replacing fuzzy membership functions in the form of first order splines by more efficient wavelets could be easily performed.

- On the practical side, on can notice from both figures 7.16 and 7.17 that human control of glycæmia injection performs quite poorly. The desired range would be, say, about

1 ± 0.3, which is far from being accessible to human control. Thus nonlinear fuzzy control design is now under progress for this application.

Chapter 8

Discussion and conclusions

In this tutorial we have discussed the wide area of nonparametric nonlinear estimation from the point of view of system identification. We have seen that a huge amount of work has been pursued in the statisticians community. We also know from numerous press releases that, in parallel, the A.I. community revitalized the same area by advertizing neural networks, fuzzy models, and neuro-fuzzy models. In addition, A.I. scientists and engineers packaged these techniques with user oriented software and even hardware. It is not until recently that the A.I. community went interested in the mathematical developments and algorithms from statistics. At the same time, statisticians became involved in the mathematical study of the methods advertized by the A.I. community and engineering practice. In parallel, the control community recognized those models and estimation algorithms as possible candidates for nonlinear black-box system identification. In this tutorial, we have tried to put together material — both classical and very modern — from different areas, and have discussed both mathematical and practical issues. Here is a summary of tentative conclusions and suggestions for future work.

Practical issues.

- *Models for prediction and simulation.* As reflected by the reported experiments, our experience has been that nonlinear nonparametric models are very good at predicting behaviours, provided that the training data set reflects all actual operating conditions that can occur. This is especially true for models that are multiresolution in nature, e.g., wavelet based models. More interesting, prediction is still efficient even for sparse training data set — a situation which is almost unavoidable for high dimensional input data. The quality of prediction can rapidly vanish outside the range of training data set, however, but this is not really surprising.

- *System monitoring and diagnostics.* The reported experiments on the gas turbine case study show that data fit is much better for our wavelet network (and even for the

neuro-fuzzy network) than for our semi-physical model. Accordingly one may expect a better performance in change detection and diagnostics by using the wavelet network. Designing a change detection procedure based on the wavelet network can be performed by applying the general asymptotic local approach discussed in [5] [90]. However since the parameters of the network have no useful interpretation, diagnostics would require learning the failure modes from training data sets : this is unrealistic since real data corresponding to failure modes are (fortunately) seldom. Thus diagnostics requires a combination of data *and* prior knowledge, preferably in the form of a (semi)-physical model : data are the current data (from safe or failured mode), and the model is used to describe prior knowledge about failure modes. In fact, gas turbine monitoring and diagnostics was successfully performed using our seemingly poor semi-physical model, see [90] [3] for an account of the results.

- *Describing prior knowledge.* Fuzzy models and their associated rules can be used to describe prior knowledge for nonlinear nonparametric models. Now if it is desired to blend the style of fuzzy rules with the mathematical quality of modern nonparametric models, we are faced with the need for a notion of "multiresolution" or "hierarchical" fuzzy rule bases. We have discussed a possible proposal toward this objective. This has to be further explored. In addition, it would be interesting to develop statistical methods checking for violation of a particular subset of fuzzy rules, this would blend methods from A.I. and statistics model based diagnostics.

- *Software support.* Our current experience can be summarized as follows. There are three different kinds of needs for nonlinear black-box identification : low dimensional input (say, $1, 2, 3$), medium dimensional input (in the range of tens), and large dimensional input (in the range of hundreds or thousands). The first case typically corresponds to curve fitting and is useful in signal or image processing, and sometimes in control. High performance algorithms based on wavelets are available today, which outperform others in both accuracy and computational cost, see Chapter 4, and softwares are available, such as C. Taswell's WavBox in Matlab language [75]. The second case has its main applications in system identification and control. There, RBF (radial basis function) networks, which provide fast noniterative training procedures, are preferred ; theoretical studies and experiments suggest that wavelet networks [89], such as discussed in this paper, are likewise more efficient candidates. Finally, sigmoid based neural networks with their iterative backpropagation algorithm, both simple and time consuming, are still effective for very large dimensional cases such as encountered in some pattern recognition applications. We have seen that alternative models with much more efficient iterative training procedures can also work well, such as Breiman's hinge functions [7] ; Breiman's hinging hyperplane algorithm fits piecewise linear models on nonlinear systems in a very efficient way.

Mathematical issues.

- *Assessing the quality of an approximation.* What is the convenient figure of merit for the estimation error $\|f - \widehat{f}\|$? We have emphasized in this tutorial the central role played by *Besov spaces*: this is a triply parametrized family of spaces of functions, that are generally smooth but may have sparse singularities. *Being smooth outside localized singularities is a common feature of most of the nonlinear systems encountered in practice, thus Besov spaces are suitable to assess the quality of an estimator.*

- *Quality of fit from noisy data, and "Cramer-Rao bounds".* Maximal risks and lower rates of convergence provide adequate frameworks; they have to be used in combination with Besov spaces. And we have shown that wavelet based estimators are optimal for systems in Besov spaces.

- *How efficient identification algorithms really are in terms of computational cost and quality of conditioning?* When orthonormal wavelet librairies can be efficiently built (this is feasible for low dimensional input, say, up to 4 or slightly more), wavelet estimators from Section 4 are the fastest ones. For very large dimensions, wavelet librairies cannot be built today, and standard sigmoid based neural networks are preferred; Breiman's hinging hyperplane models are very promising alternative candidates. In the medium range situation, wavelet networks using partial wavelet librairies seem to be efficient alternatives to RBF networks.

Research directions. Based on the material of this tutorial, we can suggest the following three major challenges for future research.

- Providing wavelet based identification methods for higher dimensional inputs. The central question here lies in the efficient construction of wavelet librairies in higher dimensions.

- Taking advantage of multiresolution in both *time and space* is a major challenge for dynamical system identification. Functional nonlinear autoregressions of the form $Y_k = f(Y_{k-1}, \ldots, Y_{k-p}) + e_k$, or their state space counterpart, are naturally used with both neural and wavelet networks. These models do not allow playing with multiresolution for time, however, since discretization is fixed and rigid. Thus a new framework would be needed for this purpose.

- Investigating the interplay between the syntax of fuzzy modelling and modern nonparametric models certainly is a topic of major practical interest. It would provide the user with ways of describing prior knowledge within nonparametric models.

Appendix A

Appendix : three methods for regressor selection

Recall that $W = \{\varphi_i \ : \ i = 1, \ldots, L\}$ is the library of the wavelet regressor candidates. Introduce the following notations.

$$v_i = \begin{bmatrix} \varphi_i(x_1) \\ \vdots \\ \varphi_i(x_N) \end{bmatrix} \tag{A.1}$$

where $\varphi_i \in W$ and $x_1, \ldots, x_N$ are input observations in the training data set

$$\mathcal{O}_1^N = \{(x_1, y_1), \ldots, (x_N, y_N)\} \ .$$

φ_i has been normalized so that v_i is unitary:

$$v_i^T v_i = 1, \quad i = 1, \ldots, L.$$

Now collect all the v_i, $i = 1, \ldots, L$ in a set V:

$$V = \{v_1, \ldots, v_L\}. \tag{A.2}$$

We also define the output observation vector

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \tag{A.3}$$

where $y_1, \ldots, y_N$ are output observations in $\mathcal{O}_1^N$.

Let $\text{span}\{v_i \ : \ i \in I\}$ be the space linearly spanned by the vectors v_i, $i \in I$, and $\mathcal{I}_M$ the set of all the M-elements subsets of the index set $\{1, 2, \ldots, L\}$. Using these notations,

selecting $I \in \mathcal{I}_M$ so that the corresponding M wavelets in W minimize the mean square residual $J(I)$ in (5.8) is equivalent to selecting the M vectors v_i from V which minimize the Euclidean distance from the vector y to the space $\text{span}\{v_i : i \in I\}$. Such an optimal solution requires an exhaustive examination of all the M-elements subsets of W, which may not be feasible in practice, because of its massive computational burden. Some sub-optimal and heuristic solutions have to be considered. In this appendix we present three heuristic procedures.

A.1 The residual based selection (RBS) : details

Define the initial residual $\gamma_0(k) = y_k$, $k = 1, \ldots, N$, with y_k the output observations in $\mathcal{O}_1^N$. Set $f_0(x) \equiv 0$.

At stage i $(i = 1, \ldots, M)$, search among W the wavelet φ_j that minimizes

$$J(\varphi_j) = \frac{1}{N} \sum_{k=1}^{N} (\gamma_{i-1}(k) - u_j \varphi_j(x_k))^2$$

where

$$u_j = \left(\sum_{k=1}^{N} (\varphi_j(x_k))^2 \right)^{-1} \sum_{k=1}^{N} \varphi_j(x_k) \gamma_{i-1}(k).$$

and $\gamma_{i-1}(k)$ $(k = 1, \ldots, N)$ are the residuals of stage $i-1$. Note

$$l_i = \arg \min_{1 \le j \le L} J(\varphi_j)$$

then φ_{l_i} is the wavelet selected at stage i. Update f_i and γ_i:

$$\begin{aligned} f_i(x) &= f_{i-1}(x) + u_{l_i} \varphi_{l_i}(x) \\ \gamma_i(k) &= \gamma_{i-1}(k) - u_{l_i} \varphi_{l_i}(x_k), \quad k = 1, \ldots, N. \end{aligned}$$

This procedure can be more conveniently described with the aid of vectorial notations as follows.

Define the initial residual vector $\gamma_0 = y$ with y as defined in (A.3) and set $f_0(x) \equiv 0$.

At stage i $(i = 1, \ldots, M)$, search among V the vector v_j that minimizes

$$J(v_j) = (\gamma_{i-1} - u_j v_j)^T (\gamma_{i-1} - u_j v_j)$$

with

$$u_j = (v_j^T v_j)^{-1} v_j^T \gamma_{i-1} = v_j^T \gamma_{i-1}$$

where the last equality is due to the normality $v_j^T v_j = 1$.

Substituting u_j into $J(v_j)$ yields

$$\begin{aligned} J(v_j) &= (\gamma_{i-1} - v_j^T \gamma_{i-1} v_j)^T (\gamma_{i-1} - v_j^T \gamma_{i-1} v_j) && \text{(A.4)} \\ &= \gamma_{i-1}^T \gamma_{i-1} + (v_j^T \gamma_{i-1})^2 v_j^T v_j - 2(v_j^T \gamma_{i-1})^2 && \text{(A.5)} \\ &= \gamma_{i-1}^T \gamma_{i-1} - (v_j^T \gamma_{i-1})^2 && \text{(A.6)} \end{aligned}$$

It turns out that minimizing $J(v_j)$ at stage i is equivalent to maximizing $(v_j^T \gamma_{i-1})^2$.

The algorithm is summarized as follows.

Regressor Selection Algorithm RBS

Step 0**:** Set $\gamma_0 = y$ and $f_0(x) \equiv 0$;

Step i **(**$i = 1, \ldots, M$**):** Let $I_i = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$, find

$$l_i = \arg \max_{j \in I_i} (v_j^T \gamma_{i-1})^2$$

and set

$$\begin{aligned} u_{l_i} &= v_{l_i}^T \gamma_{i-1} \\ f_i(x) &= f_{i-1}(x) + u_{l_i} \varphi_{l_i}(x) \\ \gamma_i &= \gamma_{i-1} - u_{l_i} v_{l_i}. \end{aligned}$$

□

It is easy to prove (see [53]):

$$\gamma_i^T \gamma_i = \gamma_{i-1}^T \gamma_{i-1} - (v_{l_i}^T \gamma_{i-1})^2$$

so $\gamma_i^T \gamma_i$ monotonically decreases as i increases. It also means that the i-th term added to $f_M(x)$ has a contribution to the minimization of $\gamma_M^T \gamma_M$ measured by $(v_{l_i}^T \gamma_{i-1})^2$.

A.2 Stepwise selection by orthogonalization (SSO) : details

At stage i of this procedure, assume that the $i-1$ already selected wavelets correspond to the vectors $v_{l_1}, \ldots, v_{l_{i-1}}$. In order to select the i-th wavelet, we have to compute the distance from y to the space $\text{span}(v_{l_1}, \ldots, v_{l_{i-1}}, v_j)$ for each $j = 1, \ldots, L$ and $j \neq l_1, \ldots, l_{i-1}$. For computational efficiency, we orthogonalize the later selected vectors v_j to the earlier selected ones. Assume that $v_{l_1}, \ldots, v_{l_{i-1}}$ are already orthonormalized and renamed as $w_{l_1}, \ldots, w_{l_{i-1}}$, then

$\text{span}(v_{l_1}, \ldots, v_{l_{i-1}}, v_j) = \text{span}(w_{l_1}, \ldots, w_{l_{i-1}}, v_j)$. For each $j = 1, \ldots, L$ and $j \neq l_1, \ldots, l_{i-1}$, compute

$$p_j = v_j - \left((v_j^T w_{l_1}) w_{l_1} + \cdots + (v_j^T w_{l_{i-1}}) w_{l_{i-1}}\right) \tag{A.7}$$

$$q_j = (p_j^T p_j)^{-\frac{1}{2}} p_j \tag{A.8}$$

then we should search the v_j, or equivalently the q_j, that minimizes

$$\begin{aligned} J(v_j) &= J(q_j) \\ &= \left[y - \left(\tilde{u}_{l_1} w_{l_1} + \cdots + \tilde{u}_{l_{i-1}} w_{l_{i-1}} + \tilde{u}_j q_j\right)\right]^T \left[y - \left(\tilde{u}_{l_1} w_{l_1} + \cdots + \tilde{u}_{l_{i-1}} w_{l_{i-1}} + \tilde{u}_j q_j\right)\right] \\ &= \left[y - W_j U_j\right]^T \left[y - W_j U_j\right] \end{aligned}$$

with the matrix $W_j = (w_{l_1}, \ldots, w_{l_{i-1}}, q_j)$ and the vector

$$U_j = (\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_{i-1}}, \tilde{u}_j)^T = \left(W_j^T W_j\right)^{-1} W_j^T y = W_j^T y \tag{A.9}$$

where the last equality is due to the orthonormality of $w_{l_1}, \ldots, w_{l_{i-1}}, q_j$. Continue the computation:

$$\begin{aligned} J(v_j) &= y^T y + U_j^T W_j^T W_j U_j - 2 U_j^T W_j^T y \\ &= y^T y + U_j^T U_j - 2 U_j^T W_j^T y \end{aligned}$$

By (A.9) we have $U_j = W_j^T y$, therefore

$$\begin{aligned} J(v_j) &= y^T y + U_j^T U_j - 2 U_j^T U_j \\ &= y^T y - U_j^T U_j \\ &= y^T y - \left(\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2 + \tilde{u}_j^2\right) \end{aligned}$$

Consequently minimizing $J(v_j)$ is equivalent to maximizing $\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2 + \tilde{u}_j^2$. By (A.9) we have

$$\begin{aligned} \tilde{u}_{l_k} &= w_{l_k}^T y, \quad k = 1, \ldots, i-1 \\ \tilde{u}_j &= q_j^T y \end{aligned}$$

so $\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2$ is independent of q_j. We conclude that minimizing $J(v_j)$ is equivalent to maximizing $\tilde{u}_j^2 = (q_j^T y)^2$.

After M iterations, the values of $l_1, \ldots, l_M$ are determined, as well as $\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}$. We still need to determine the values of $u_{l_1}, \ldots, u_{l_M}$ in

$$f_M(x) = \sum_{i=1}^{M} u_{l_i} \varphi_{l_i}(x).$$

By the definitions of w_l and $\tilde{u}_l$,

$$y = [w_{l_1}, \ldots, w_{l_M}][\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T + \gamma_M$$

on the other hand,

$$y = [v_{l_1}, \ldots, v_{l_M}][u_{l_1}, \ldots, u_{l_M}]^T + \gamma_M$$

therefore,

$$[w_{l_1}, \ldots, w_{l_M}][\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T = [v_{l_1}, \ldots, v_{l_M}][u_{l_1}, \ldots, u_{l_M}]^T. \quad \text{(A.10)}$$

In (A.7) and (A.8) let $j = l_i$, then combining them yields

$$\begin{aligned} v_{l_i} &= \left((v_{l_i}^T w_{l_1})w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}})w_{l_{i-1}}\right) + p_{l_i} \\ &= \left((v_{l_i}^T w_{l_1})w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}})w_{l_{i-1}}\right) + (p_{l_i}^T p_{l_i})^{\frac{1}{2}} q_{l_i} \\ &= \left((v_{l_i}^T w_{l_1})w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}})w_{l_{i-1}}\right) + (p_{l_i}^T p_{l_i})^{\frac{1}{2}} w_{l_i} \\ &= \left(\alpha_{1i} w_{l_1} + \cdots + \alpha_{i-1i} w_{l_{i-1}}\right) + \alpha_{ii} w_{l_i} \end{aligned}$$

with

$$\begin{aligned} \alpha_{ki} &= v_{l_i}^T w_{l_k}, \quad k = 1, \ldots, i-1 \\ \alpha_{ii} &= (p_{l_i}^T p_{l_i})^{\frac{1}{2}} \end{aligned}$$

Consequently

$$[w_{l_1}, \ldots, w_{l_M}]A = [v_{l_1}, \ldots, v_{l_M}] \quad \text{(A.11)}$$

where A is the triangular matrix

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & & \alpha_{1M} \\ 0 & \alpha_{22} & \alpha_{23} & \cdots & & \alpha_{2M} \\ 0 & 0 & \alpha_{33} & \cdots & & \alpha_{3M} \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & \alpha_{M-1M-1} & \alpha_{M-1M} \\ 0 & 0 & \cdots & & 0 & \alpha_{MM} \end{bmatrix}$$

Then, u_{l_i} can be obtained by solving the triangular system of equations obtained by combining (A.10) and (A.11):

$$A\,[u_{l_1}, \ldots, u_{l_M}]^T = [\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T. \quad \text{(A.12)}$$

Let us summarize the algorithm as follows.

Regressor Selection Algorithm SSO

Step 1**:** find

$$l_1 = \arg \max_{1 \le j \le L} \left(v_j^T y\right)^2$$

set

$$\begin{aligned} \tilde{u}_{l_1} &= v_{l_1}^T y \\ w_{l_1} &= v_{l_1} \\ \alpha_{11} &= 1 \end{aligned}$$

Step i **(**$i = 2, \ldots, M$**):** Let $I_i = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$. For each $j \in I_i$, compute

$$\begin{aligned} p_j &= v_j - \left((v_j^T w_{l_1}) w_{l_1} + \cdots + (v_j^T w_{l_{i-1}}) w_{l_{i-1}}\right) \\ q_j &= (p_j^T p_j)^{-\frac{1}{2}} p_j \end{aligned}$$

find

$$l_i = \arg \max_{j \in I_i} \left(q_j^T y\right)^2$$

and set

$$\begin{aligned} \tilde{u}_{l_i} &= q_{l_i}^T y \\ w_{l_i} &= q_{l_i} \\ \alpha_{ki} &= v_{l_i}^T w_{l_k}, \quad k = 1, \ldots, i-1 \\ \alpha_{ii} &= (p_{l_i}^T p_{l_i})^{\frac{1}{2}} \end{aligned}$$

Step $M+1$**:** solve (A.12) to obtain u_{l_i}, $i = 1, \ldots, M$, and build

$$f_M(x) = \sum_{i=1}^{M} u_{l_i} \varphi_{l_i}(x).$$

□

A.3 Backward elimination (BE) : details

The regression with all the wavelets of W is written as

$$f_L(x) = \sum_{i=1}^{L} u_i \varphi_i(x)$$

where u_i are determined by the least squares algorithm:

$$(u_1, \ldots, u_L)^T = \left[(v_1, \ldots, v_L)^T (v_1, \ldots, v_L)\right]^{-1} (v_1, \ldots, v_L)^T y. \tag{A.13}$$

Note that inverting the matrix $(v_1, \ldots, v_L)^T (v_1, \ldots, v_L)$ may cause problem when it is singular. This situation rarely occurs with the set V of vectors corresponding to the wavelet library W. Whenever it happens, the two previously presented regressor selection algorithms should be used.

The residuals

$$\gamma_L(k) = y_k - f_L(x_k), \quad k = 1, \ldots, N$$

can be written in its vectorial form as:

$$\gamma_L = y - (v_1, \ldots, v_L)(u_1, \ldots, u_L)^T. \tag{A.14}$$

Combining (A.13) and (A.14) we get

$$\gamma_L^T \gamma_L = y^T y^T - y^T V_0 (V_0^T V_0)^{-1} V_0^T y$$

where the matrix $V_0 = (v_1, \ldots, v_L)$.

If we remove one wavelet, say φ_j from $f_L(x)$, the same computation can be repeated to get a similar result

$$\gamma_{L-1}^T \gamma_{L-1} = y^T y^T - y^T C(v_j|V_0) \left(C(v_j|V_0)^T C(v_j|V_0)\right)^{-1} C(v_j|V_0)^T y$$

where the operator C means the complement of a matrix, i.e., if a matrix $U = [U_1, U_2, U_3]$, then $C(U_2|U) = [U_1, U_3]$. Hence, the increment of the sum of square residual caused by removing φ_j from $f_L(x)$ is

$$\begin{aligned} J(\varphi_j) &= \gamma_{L-1}^T \gamma_{L-1} - \gamma_L^T \gamma_L \\ &= y^T V_0 (V_0^T V_0)^{-1} V_0^T y - y^T C(v_j|V_0) \left(C(v_j|V_0)^T C(v_j|V_0)\right)^{-1} C(v_j|V_0)^T y \end{aligned} \tag{A.15}$$

Removing from $f_L(x)$ the wavelet φ_j that minimizes (A.15) yields $f_{L-1}(x)$. Repeat the same procedure to remove another wavelet from $f_{L-1}(x)$, and so on. This results in the following algorithm.

Regressor Selection Algorithm BE$_{\text{full}}$

Step 0**:** set $V_0 = (v_1, \ldots, v_L)$;

Step i $(i = 1, \ldots, L-M)$**:** let $I_i = \{j : \ j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$, find

$$l_i = \arg\max_{j \in I_i} y^T C(v_j|V_{i-1}) \left(C(v_j|V_{i-1})^T C(v_j|V_{i-1})\right)^{-1} C(v_j|V_{i-1})^T y$$

set $V_i = C(v_{l_i}|V_{i-1})$;

Step $L-M+1$**:** let $I_{L-M+1} = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{L-M}\}$, build

$$f_M(x) = \sum_{j \in I_{L-M+1}} u_j \varphi_j(x)$$

with u_j the components of the vector u given by

$$u = \left(V_{L-M}^T V_{L-M}\right)^{-1} V_{L-M}^T \, y.$$

□

The computation required by this procedure is quite heavy. For instance $L-i+1$ matrices need to be inverted at step i. The computation for inverting the matrices can be reduced in the following way.

For any matrix $U = [U_1, U_2, U_3]$ where U_1, U_2, U_3 are sub-blocs of U,

$$U^T U = \begin{bmatrix} U_1^T U_1 & U_1^T U_2 & U_1^T U_3 \\ U_2^T U_1 & U_2^T U_2 & U_2^T U_3 \\ U_3^T U_1 & U_3^T U_2 & U_3^T U_3 \end{bmatrix}.$$

Assume $\left(U^T U\right)^{-1}$ is already calculated and partitioned in the same way as $U^T U$:

$$\left(U^T U\right)^{-1} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \Lambda_{13} \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} \\ \Lambda_{31} & \Lambda_{32} & \Lambda_{33} \end{bmatrix}$$

Then the following formula can be easily verified:

$$\begin{aligned} \left([U_1, U_3]^T [U_1, U_3]\right)^{-1} &= \begin{bmatrix} U_1^T U_1 & U_1^T U_3 \\ U_3^T U_1 & U_3^T U_3 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \Lambda_{11} & \Lambda_{13} \\ \Lambda_{31} & \Lambda_{33} \end{bmatrix} - \Lambda_{22}^{-1} \begin{bmatrix} \Lambda_{12} \\ \Lambda_{32} \end{bmatrix} \begin{bmatrix} \Lambda_{21} & \Lambda_{23} \end{bmatrix} \end{aligned} \tag{A.16}$$

In this way only $V_0^T V_0$ needs to be actually inverted with conventional method. Using(A.16), $\left(C(v_j|V_i)^T C(v_j|V_i)\right)^{-1}$ can be obtained from sub-blocs of $\left(V_i^T V_i\right)^{-1}$.

This procedure can be further simplified as follows.

Assume that $f_L(x)$ is built with all the wavelets of W as before. Now eliminate one wavelet from $f_L(x)$, say φ_j, but keep the values of u_l unchanged, $l = 1, \ldots, L$, the residual becomes

$$\gamma_{L-1}(k) = y_k - (f_L(x_k) - u_j \varphi_j(x_k)) = \gamma_L(k) + u_j \varphi_j(x_k), \quad k = 1, \ldots, N.$$

so

$$\gamma_{L-1} = \gamma_L + u_j v_j$$

Then

$$\begin{aligned}\gamma_{L-1}^T\gamma_{L-1} &= \gamma_L^T\gamma_L + u_j^2 v_j^T v_j + 2u_j\gamma_L^T v_j \\ &= \gamma_L^T\gamma_L + u_j^2 + 2u_j\gamma_L^T v_j\end{aligned}$$

The last term of this equation can be neglected under the assumptions that γ_L is close to zero mean and independent of v_i. Therefore

$$\gamma_{L-1}^T\gamma_{L-1} - \gamma_L^T\gamma_L \approx u_j^2$$

This means that removing φ_j from $f_L(x)$ will cause a increment of the sum of square residuals approximatively equal to u_j^2. Repeating the same reasoning on $f_{L-1}(x)$, $f_{L-2}(x)$, etc. yields the following procedure.

Regressor Selection Algorithm BE

Step 0**:** set $V_0 = (v_1, \ldots, v_L)$;

Step i $(i = 1, \ldots, L-M)$**:** let $I_i = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$ and compute

$$u = \left(V_{i-1}^T V_{i-1}\right)^{-1} V_{i-1}^T y$$

where u is a vector composed of u_j, $j \in I_i$;
find

$$l_i = \arg\min_{j \in I_i} u_j^2$$

set $V_i = C(v_j | V_{i-1})$;

Step $L-M+1$**:** let $I_{L-M+1} = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{L-M}\}$, build

$$f_M(x) = \sum_{j \in I_{L-M+1}} u_j \varphi_j(x)$$

with u_j the components of the vector u given by

$$u = \left(V_{L-M}^T V_{L-M}\right)^{-1} V_{L-M}^T\, y.$$

□

Note that equation (A.16) is used for inverting $V_i^T V_i$, $i > 0$, only $V_0^T V_0$ is inverted with conventional algorithm. Alternatively, if the mother wavelet function φ is chosen to have compact support, then the matrices V_i and $V_i^T V_i$ are sparse. $V_i^T V_i$ is symmetric and usually has diagonal dominance. In such situations, and for large matrices $V_i^T V_i$, instead of directly computing

$$u = \left(V_i^T V_i\right)^{-1} V_i^T y$$

iterative methods [84] should be used for solving

$$\left(V_i^T V_i\right) u = V_i^T y.$$

The above proposed algorithms RBS, SSO and BE have been implemented in Matlab 4.1 language. Algorithm $\mathrm{BE}_{\mathrm{full}}$ is not implemented due to its high computational cost.

Bibliography

[1] H. Akaike, *Statistical predictor identification*, Ann. Inst. Math. Statist., 22 (1970), pp. 203–217.

[2] A. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. on Information Theory, 39 (1993).

[3] M. Basseville, A. Benveniste, G. Mathis, and Q. Zhang, *Monitoring the combustion set of a gas turbine*, in Proceedings of SAFEPROCESS'94, 1994, p. .

[4] A. Benveniste, *Digital Signal Processing Techniques and Applications*, Advances in Control and Dynamic Systems, Academic Press, 1993, ch. Multiscale signal Processing : from QMF to Wavelets.

[5] A. Benveniste, M. Basseville, and G. Moustakides, *The asymptotic local approach to change detection and model validation*, IEEE Trans. on Automatic Control, 32 (1987), pp. 583–592.

[6] O. Besov, *On a family of functional spaces: embedding theorems and applications*, Doklady Acad. Nauk SSSR, 126, 1163-1165 (1959).

[7] L. Breiman, *Hinging hyperplanes for regression, classification and function approximation*, IEEE Trans. on Information Theory, 39 (1993), pp. 999–1013.

[8] L. Breiman, J. Friedman, J. Olshen, and C. Stone, *Classification and regression trees*, Wadsworth, Belmont, California, 1984.

[9] N. Cencov, *Statistical decision rules and optimal inference*, Amer. Math. Soc. Transl., 53 (1982). Providence, R.I.

[10] S. Chen, S. Billings, and W. Luo, *Orthogonal least squares methods and their application to non-linear system identification*, Int. J. Control, 50 (1989), pp. 1873–1896.

[11] S. Chen, C. Cowan, and P. Grant, *Orthogonal least squares learning algorithm for radial basis function networks*, IEEE Trans. on Neural Networks, 2 (1991), pp. 302–309.

[12] P. Craven and G. Wahba, *Smoothing noisy data with spline functions*, Numer. Math., 31 (1979), pp. 337–403.

[13] S. Csibi, *Stochastic Processes with Learning Properties*, Springer-Verlag, Berlin, 1975.

[14] I. Daubechies, *Ten lectures on wavelets*, CBMS-NSF regional series in applied mathematics, CBMS-NSF regional conference, 1992.

[15] Delyon and A. Juditsky, *Optimal Estimators for Functional Autoregression*, Tech. Rep. IRISA, in preparation, 1994.

[16] B. Delyon, *Orthogonal and Biorthogonal Wavelets*, Technical Report IRISA, 732, 1993.

[17] B. Delyon and A. Juditsky, *Wavelet Estimators, Global Error Measures Revisited*, Technical Report 782, IRISA, 1993.

[18] B. Delyon, A. Juditsky, and A. Benveniste, *Accuracy analysis for wavelet networks*, IEEE Transactions on neural networks, (1994). to appear.

[19] R. DeVore, B. Jawerth, and V. Popov, *Compression of wavelet decompositions*, Amer. J. Math., To appear (1994).

[20] L. Devroye, *Any discrimination rule can have an arbitrary bad probability of error for final sample size*, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-4 (1982), pp. 154–157.

[21] L. Devroye and L. Györfi, *Nonparametric Density Estimation L_1 View*, J. Wiley, New-York, 1985.

[22] L. Devroye and T. Wagner, *Distribution free consistency result in nonparametric discrimination and regression function estimation*, The Annals of Statistics, 8 (1980), pp. 231–239.

[23] D. Donoho, *Interpolating Wavelet Transforms*, Technical Report, Department of Statistics, Stanford University `ftp playfair.stanford.edu`, 1993.

[24] ——, *Smooth Wavelet Decompositions with blocky Coefficient Kernels*, Technical Report, Department of Statistics, Stanford University `ftp playfair.stanford.edu`, 1993.

[25] D. Donoho and I. Johnstone, *Minimax Estimation via Wavelet Shrinkage*, Technical Report, Department of Statistics, Stanford University `ftp playfair.stanford.edu`, 1992.

[26] ——, *Minimax Risk over l_p-balls*, Technical Report, Department of Statistics, Stanford University `ftp playfair.stanford.edu`, 1992.

[27] ——, *Adapting to Unknown Smoothness via Wavelet Shrinkage*, Technical Report, Department of Statistics, Stanford University, `ftp playfair.stanford.edu`, 1993.

[28] D. Donoho, I. Johnstone, G. Kerkyacharian, and D. Picard, *Density estimation by wavelet thresholding*, Technical Report, Department of Statistics, Stanford University `ftp playfair.stanford.edu`, 1993.

[29] ——, *Wavelet Shrinkage: Asymptopia*, Manuscript on `ftp playfair.stanford.edu`, 1993.

[30] N. Draper and H. Smith, *Applied regression analysis*, Series in Probability and Mathematical Statistics, Wiley, 1981. Second edition.

[31] D. Dubois and H. Prade, *Conditional Logic in Expert Systems*, Elsevier Science Publishers B.V., North Holland, 1991, ch. Conditioning, non-monotonic logic, and non-standard uncertainty models, pp. 115–158.

[32] ——, *Fuzzy sets in approximate reasoning, part 1*, Fuzzy Sets and Systems, 40 n°1 (1992).

[33] M. Duflo, *Recursive Stochastic Methods*, Springer-Verlag, Berlin, 1993.

[34] S. Efroimovich and M. Pinsker, *Estimation of square-integrable spectral density based on a sequence of observations*, Problems of Information Transmission (in Russian), pp. 182-196 (1982).

[35] ——, *Estimation of square-integrable probability density of a random variable*, Problems of Information Transmission (in Russian), pp. 175-189 (1983).

[36] ——, *A learning algorithm for nonparametric filtering*, Automatika i Telemehanika (in Russian), 11, 58-65 (1984).

[37] J. Friedman, *Multivariate adaptive regression splines (with discussion)*, The Annals of Statistics, 19 (1991), pp. 1–141.

[38] J. Friedman and W. Stuetzle, *Projection pursuit regression*, J. Amer. Stat. Assoc., 76 (1981), pp. 817–823.

[39] P. Glorennec, *A general class of fuzzy inference systems*, in Proc. of CES2 Conf., Prague, 1993.

[40] W. Härdle and J. Marron, *Optimal bandwidth selection in nonparametric regression function estimation*, The Annals of Statistics, 13 (1985), pp. 1465–1481.

[41] P. Huber, *Projection pursuit (with discussion)*, The Annals of Statistics, 13 (1985), pp. 435–475.

[42] K. Hunt, D. Sbarbaro, R. Zbikowski, and P. Gawthrop, *Neural networks for control systems — a survey*, Automatica, 28 n°6 (1992), pp. 1083–1112.

[43] I. Ibragimov and R. Khasminskij, *Statistical Estimation Asymptotic Theory*, Springer-Verlag, Berlin, 1981.

[44] S. Jaffard and P. Laurentçot, *Wavelets : A Tutorial*, Academic Press, 1989, ch. Wavelets and P.D.E.'s.

[45] A. Juditsky, *Adaptive Wavelet Estimators*, Technical Report 815, IRISA, 1994.

[46] G. Kerkyacharian and D. Picard, *Density estimation in besov spaces*, Stat. and Prob. Letters, 13, 15-24 (1992).

[47] A. Korostelev and A. Tsybakov, *Minimax Theory of Image Reconstruction*, Springer-Verlag, Berlin, 1981.

[48] C. Lee, *Fuzzy logic in control systems, parts 1 and 2*, IEEE Trans. on Systems, Man, and Cybernetics, 20 n°2 (1990).

[49] K. Li, *Asymptotic optimality of c_L and generalized cross-validation in ridge regression and application to the spline smoothing*, The Annals of Statistics, 14 (1986), pp. 1101–1112.

[50] ——, *Asymptotic optimality of c_L and generalized cross-validation : discrete index set*, The Annals of Statistics, 15 (1987), pp. 958–975.

[51] L. Ljung, *Perspectives on the process of identification*, in Proceedings of the 12th IFAC World Congress, Sydney, 1993.

[52] ——, *Neural networks in identification, a tutorial*, in Proc. of the 10th IFAC Symposium on Identification and System Parameter Estimation, Kopenhagen, July 4–6, 1994.

[53] S. Mallat and Z. Zhang, *Matching pursuit with time-frequency dictionaries*, Technical Report 619, New-York University, Computer Science Department, Aug. 1993.

[54] C. Mallows, *Statistical predictor identification*, Technometrics, 15 (1973), pp. 661–675.

[55] Y. Meyer, *Ondelettes et Opérateurs*, Hermann, 1990.

[56] J. Morgan and J. Sonquist, *Problems in the analysis of survey data, and a proposal*, J. Amer. Stat. Assoc., 58 (1963), pp. 415–434.

[57] H. Müller and U. Stadtmüller, *Variable bandwidth kernel estimators of regression curves*, The Annals of Statistics, 15(1), 182–201 (1987).

[58] E. Nadaraya, *On estimating regression*, Theory of Prob. and Appl., 9 (1964), pp. 141–142.

[59] K. Narendra and K. Parthasarathy, *Identification and control of dynamical systems using neural networks*, IEEE Trans. on Neural Networks, 1 n^{o}1 (1990), pp. 4–27.

[60] A. Nemirovskij, *Nonparametric estimation of smooth regression functions*, Izv. Acad. Nauk SSSR, Techn. Kibern. (in Russian), 3, 50-60 (1985).

[61] G. Oppenheim and B. Portier, *Commande adaptative du processus de Markov* $x_{t+1} = f_t + u_t + x_t,\ \ t \in N$, Technical Report 90–18, Université d'Orsay, 1990.

[62] E. Parzen, *On estimation of probability density function and the mode*, Ann. of Math. Stat., 33 (1962), pp. 1065–1076.

[63] P. Petrushev and V. Popov, *Rational Approximation Of Real Functions*, Cambridge University Press, Cambridge, 1987.

[64] G. Plotkin, *A Structural Approach to operational Semantics*, Lect. Notes, Aarhus Univ, 1981.

[65] T. Poggio and F. Girosi, *Networks for approximation and learning*, Proceedings of the IEEE, 78 (1990), pp. 1481–1497.

[66] B. Polyak and A. Tsybakov, *Asymptotical optimality of c_p criterion for projection regression estimates*, Theory of Prob. and Appl., 35 (1990), pp. 305–317.

[67] B. Portier, *Estimation non paramétrique et commande adaptative de processus Markoviens non linéaires*, PhD thesis, Université Paris Sud, Orsay, 1992.

[68] S. Qian and D. Chen, *Signal representation using adaptive normalized gaussian functions*, Signal Processing, 36 (1994).

[69] J. Rice, *Bandwidth choice for nonparametric regression*, The Annals of Statistics, 12 (1984), pp. 1215–1230.

[70] M. Rosenblatt, *Remarks on some nonparametric estimates of density functions*, Ann. of Math. Stat., 27 (1956), pp. 832–835.

[71] ——, *Curve estimation*, Ann. of Math. Stat., 42 (1971), pp. 1815–1842.

[72] W. Sickel, *Spline representations of functions in besov-triebel-lizorkin spaces on* $\mathbf{R}^n$, Forum Math., 2, 451-476 (1990).

[73] E. Sontag, *Nonlinear regulation : the piecewise linear approach*, IEEE Trans. on Automatic Control, 26 (1981), pp. 346–358.

[74] C. Stone, *Optimal global rates of convergence for nonparametric regression*, The Annals of Statistics, 10 (1982), pp. 1040–1053.

[75] C. Taswell, *WavBox*, Public domain Matlab toolbox Anonymous ftp: simplicity.stanford.edu : /pub/taswell, 1993.

[76] H. Triebel, *Theory of Function Spaces*, Birkhäuser Verlag, Berlin, 1983.

[77] ——, *Theory of Function Spaces II*, Birkhäuser Verlag, Berlin, 1993.

[78] P. Vaidyanathan, *Quadrature mirror filters banks, m-band extensions and perfect reconstruction techniques*, IEEE-ASSP Magazine, 4 (1987), pp. 4–20.

[79] J. Van Ryzin, *Bayes risk consistency of classification procedures using density estimation*, Sankhya, 28 (1966), pp. 261–270.

[80] G. Wahba, *Spline functions for observational data*, SIAM, Philadelphia, PA, 1991.

[81] L. Wang, *Fuzzy systems are universal approximators*, in Proc. First IEEE Conf. on Fuzzy Systems, 1163-1169, San Diego, 1992.

[82] G. Watson, *Smooth regression analysis*, Sankhya, Series, A (1969), pp. 359–372.

[83] C. Wolverton and T. Wagner, *Asymptotically optimal discriminant functions for pattern classifications*, IEEE Trans. on Information Theory, IT-15 (1969), pp. 258–265.

[84] D. M. Young, *Iterative solution of large linear systems*, Academic Press, 1971.

[85] L. Zadeh, *Fuzzy logic, neural networks, and soft computing*, Communications of the ACM, 37 n^{o}3 (March 1994), pp. 77–86.

[86] Q. Zhang, *Contribution à la surveillance de procédés industriels*, PhD thesis, Université de Rennes I, Dec. 1991.

[87] ——, *Wavelet networks : the radial structure and an efficient initialization procedure*, Technical Report LiTH-ISY-I-1423, Linköping University, Oct. 1992.

[88] ——, *Regressor selection and wavelet network construction*, Technical Report 709, Inria, Apr. 1993.

[89] Q. Zhang, *WaveNet*, Public domain Matlab toolbox Anonymous ftp: ftp.irisa.fr : /local/wavenet, 1993.

[90] Q. Zhang, M. Basseville, and A. Benveniste, *Early warning of slight changes in systems and plants with application to condition based maintenance*, Automatica, 30 (1994), pp. 95–113. Special Issue on Statistical Methods in Signal Processing and Control.

[91] Q. Zhang and A. Benveniste, *Wavelet networks*, IEEE Trans. on Neural Networks, 3 (1992), pp. 889–898.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399